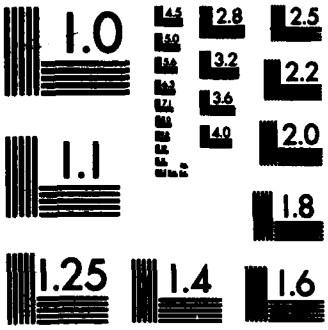
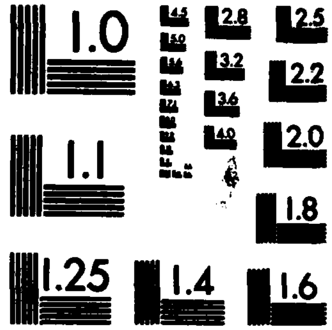
AD-A120 477 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) (U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA WATERWHEEL PROGRAM OFFICE R W COVEY ET AL. 15 SEP 82 1/4

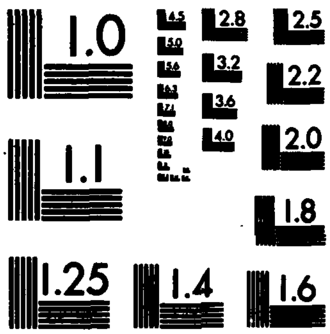
UNCLASSIFIED N00014-81-C-0715 F/G 15/3 NL



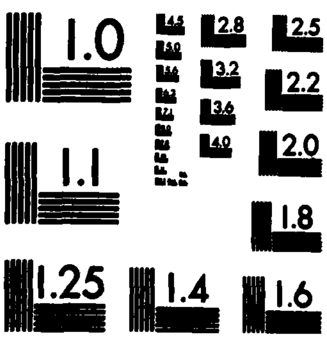
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



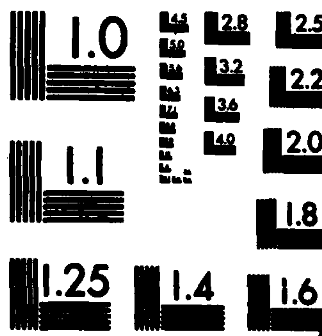
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

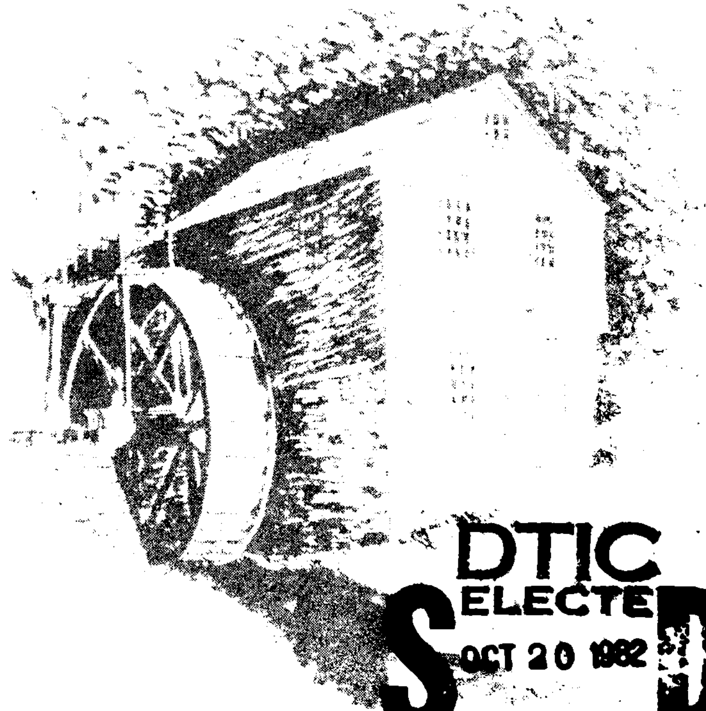
SSP-GOO-OG-0049-82
15 SEPTEMBER 1982

UNCLASSIFIED

DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION

NADS

Contract N00014-81-C-0715



SPECIAL PROGRAMS

TECHNICAL REPORT

PREPARED FOR
OFFICE OF THE CHIEF OF NAVAL OPERATIONS
OP-654

TRW
DEFENSE SYSTEMS GROUP
WATERWHEEL PROGRAM OFFICE
7600 COLSHIRE DRIVE • MCLEAN, VIRGINIA 22101

SSP-G00-0G-0049-82

DESIGN NOTEBOOK
FOR
NAVAL AIR DEFENSE SIMULATION (NADS)

FINAL REPORT

15 September 1982

Prepared by:

R.W. Covey
R.F. Evans
K.F. Nelson
R.V. Powers

Submitted by:

R. Anspach

Contract N00014-81-C-0715

TRW
WATERWHEEL PROGRAM OFFICE

FOREWORD

This document is a revision of three previously published reports that documented the Naval Air Defense Simulation model, NADS. This revision is the Final Report of the software development effort under ONR contract N00014-81-C-0715, performed by the Waterwheel Program Office of the TRW Defense Systems Group, for the Chief of Naval Operations, OP-654.

This document replaces the following three documents that were originally prepared for the Defense Nuclear Agency under contract DNA001-79-C-0242 in March 1981:

1. Design Notebook for NADS, Unclassified, TRW document No.SSP-M00-0003-81, and subsequent revision pages dated 9/21/81.

2. Dictionary of Variables for NADS, Unclassified, TRW document No. SSP-M00-OG-0005-81, and subsequent revision pages dated 9/21/81.

3. Maintenance Data for NADS, Unclassified, TRW document No. SSP-M00-OG-0006-81.

PREFACE

The Naval Air Defense Simulation (NADS) is a large scale simulation of the defenses of a carrier battle group under attack by antiship missiles launched from ships, submarines, and bombers. NADS treats in considerable detail, the airborne assets of the attacking Red force and the AAW assets of the Blue defending force. A prominent feature of NADS is its simulation of the CV Battle Group's acquisition of tactical information by its own resources, supplemented by external surveillance information. The battle group command center maintains a continually updated "Blue Perception" of the tactical situation. All of the tactical decisions are based on that perceived picture invoked by data that are often deficient in accuracy, completeness, and timeliness.

The NADS model is organized to fit the conventional defense-in-depth zones; the outer air battle, the SAM area defense, and the terminal defenses. The Red attack force comprises bombers, escort fighters, recon aircraft, standoff jammers, and antiship missiles that are launched by the bombers or by ships and submarines. (The Red ships and submarines are not simulated.)

The outer air battle is conducted by interceptors on combat air patrol stations (CAP) and deck-launched interceptors (DLI) from the carriers. Coordinated by air controllers aboard ships or early warning aircraft, the interceptors normally are assigned targets by the command center but can select their own targets if communications are disrupted by jamming or battle damage. Their objective is to intercept the bombers before their launching of antiship missiles. The interceptors will engage the Red missiles when feasible, but in most instances an antiship missile that is successfully launched will penetrate to the SAM area defense zone.

The SAM area defenses are conducted by AAW escort ships employing conventional or nuclear surface-to-air missiles. Coordination is by the same command center that coordinates the outer air battle. The ships are characterized by their radar performance, number and type of missile launchers, number of fire control channels, number and types of missiles, and their tactical data net capability. The command center assigns specific targets to individual ships, as well as assigning sectors to be covered in the absence of target assignments. Each ship manages the commitment and tie-up times of its principal subsystems in the attempt to engage all targets. Targets that the SAM systems miss or cannot engage are passed on to the terminal defenses of the targeted ships.

The terminal defense phase, which is simulated in much less detail than the preceding higher-payoff phases, is used in NADS to establish the number of hits for the case of conventional warheads, or the time and position of the bursts for the nuclar warhead case.

Nuclear weapons effects are computed for each ship, aircraft, and missile in the proximity of each nuclear burst. Damage is scored by comparing the computed levels with the input vulnerability threshold values for each type of potential victim.

The overall theme of the NADS simultion is to quantify the net performance of the defensive systems when their potential capabilities are limited by imperfect tactical information. Because of incomplete and delayed information, in NADS as in actual combat, some targets are overengaged while others are unengaged, interceptors are placed on CAP stations too late or too soon, the stations are imperfectly placed, interceptors are sometimes assigned to fighters instead of bombers, and the launching of DLI is not ideally timed. NADS was specifically designed to facilitate identification of the critical deficiencies and to evaluate the net worth of prospective ways to minimize their impact. The software has been designed in a modular form so that the effects of changes in hardware characteristics and in tactical logic can be readily examined without extensive reprogramming.

NADS was initially developed by Project Waterwheel of the TRW Defense and Space Systems Group under contract to The Defense Nuclear Agency, and in coordination with the Office of the Chief of Naval Operations, OP-654. OP-65 and other Navy offices have sponsored additional development.

This document is one of three that comprise the complete documentation of the NADS simulation:

NADS DESIGN NOTEBOOK

A comprehensive description of the design of NADS, covering the tactical activities and physical phenomena that are being simulated and the design of the specific software elements that are used to perform the simulations. The design rationale and compromises are described, using the diagrams and flowcharts that led to the coding of each module and subroutine. The information that was formerly covered by the publications "Maintenance Data for NADS" and "Dictionary of Variables for NADS" is now included in the Design Notebook.

NADS USERS' MANUAL

User-oriented information on computer facility requirements, input data, input formats, user-constructed data files, and operating instructions to perform NADS simulations in a stand-alone mode or in concert with other models (e.g., NNWS).

NADS SOURCE CODE

The complete source code listings in GPSS and FORTRAN for all of the elements of NADS, extensively commented.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont.)

LIST OF ILLUSTRATIONS (Cont.)

(rev. 10/15/82)

LIST OF ILLUSTRATIONS (Cont.)

1. DESCRIPTIVE OVERVIEW

1.1 GENERAL

NADS is designed to apply the defense-in-depth concept providing multiple opportunities to intercept an attacking unit before it reaches the vital area (Battle Group center). It employs combat air patrol (CAP) aircraft, deck-launched intrceptors (DLI), long range surface-to-air missiles (SAM), medium range SAMs, and terminal defense weapons in subsequent order, to destroy the threat prior to impact in the vital area. The specific deployment of individual forces is variable to fit any particular combination of threat, available resources, mission, and geographical location.

The Red attack is composed of aircraft and antiship missiles. Four types of aircraft are modelled - bombers, reconnaissance, standoff jammers, and fighter escorts. Missiles may be air-launched, submarine-launched, or surface ship launched. Air-launched missiles are carried by bombers only. The submarine and surface ship missiles ar treated jointly as sea-launched missile types.

1.2 BLUE FORCE COMPOSITION

The simulation incorporates a Carrier Battle Group (CVBG) consisting of up to five aircraft carriers supported by multiple AAW and ASW ships. These ships along with aircraft from the carriers make up the AAW force. Operations of the ASW and surface warfare elements are not modelled explicitly in NADS, but non-AAW ships can be included as potential targets of the attack.

The major elements modelled are shown in Figure 1-1. Although the Officer in Tactical Command (OTC) is supported by coordinators responsible for the various specialized defense operations, they are not individually modelled in NADS. Figure 1-1 shows the primary elements that support the Antiair Warfare Coordinator (AAWC); the capabilities and responsibilities that are modelled for each are discussed in the following paragraphs.

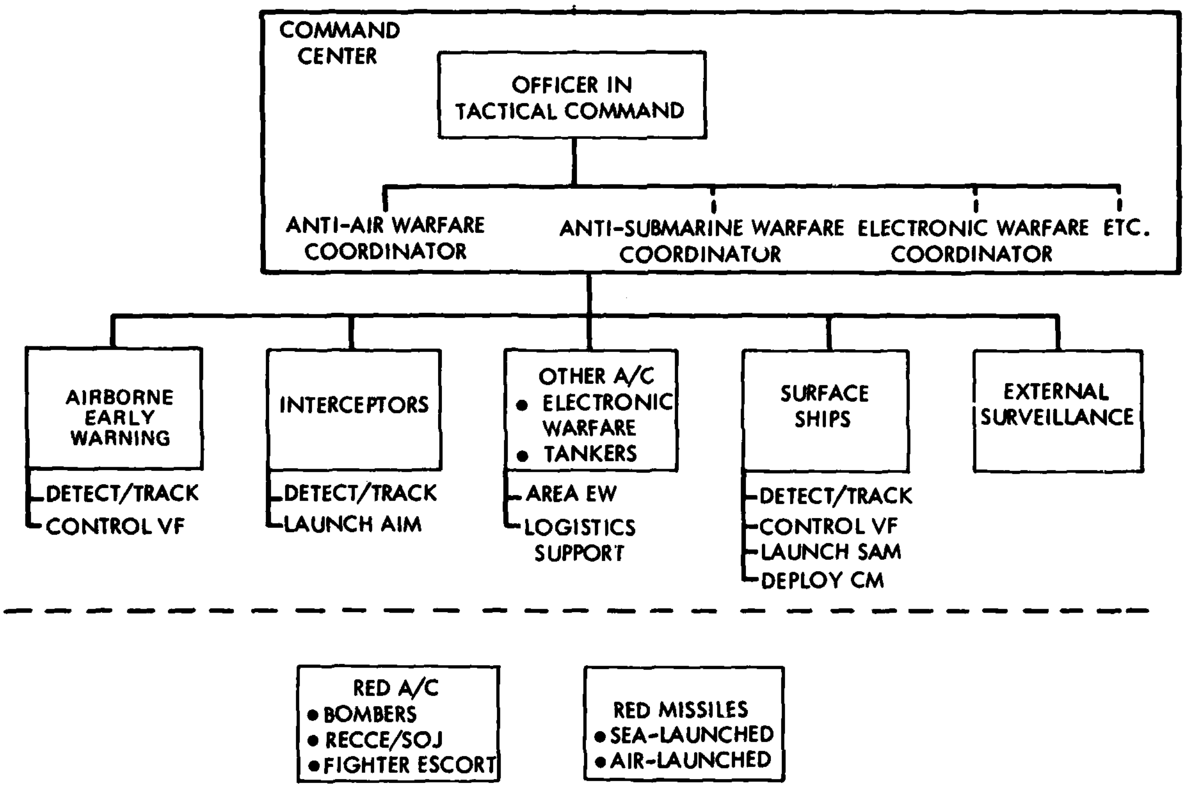


Figure 1-1. Major AAW Elements Modelled

Command Center

The CVBG command and control functions are aggregated into a single command center. The functions include those performed by the OTC, AAWC, and any interactions with other coordinators. The command center has access to information from messages, data networks, and external surveillance. It provides high level decision making and coordination among the elements of the defending force. A more detailed description of the command center functions is given in Section 1.5, Command and Control.

Surface Ships

The surface ships are modelled to detect, track, and attack airborne threats. The number of launchers, tracking and guidance capacity, and number and type of missiles carried varies among ships. Therefore the simulation allows the user to select from ships of all major classes and to dispose of them in any desired formation. SAM ships will generally be stationed near the vital area to be defended. Surface picket ships can be placed a large distance from the vital area to extend the range of detection and allow for early interception of threats, if they appear on that particular axis. Picket ships also have the ability to control airborne interceptors in attacks on enemy aircraft. The aircraft carriers, usually the protected units in the force, may also have SAM and terminal defense capability. NADS does not restrict the characteristics of ships to existing classes.

The ship model treats the following components:

Air search radar

Radar tracking capacity

SAM fire control systems

SAM launching systems

SAM stores

Decision making

Defense countermeasures

Interceptor vectoring capability

Tactical data system

Communications

The level of detail to which each component is modelled varies in proportion to the component's relative influence on the total ship performance. Various ship types are accommodated by using different numeric values to characterize these components.

The sequence of events involving these components is described in simplified form in Figure 1-2. Detections are made on radars or passive detection systems. Radar detections develop target tracks that enable the decision maker to engage the track by use of SAMs, interceptors, countermeassures, or combinations of each. When a SAM launch is chosen, the target track is assigned to a fir control channel, and an appropriate SAM is selected for launching. After a SAM is launched, the fire control channel remains occupied until after the intercept. The ship will use data from the tactical data system and in turn contribute to it. The decision making function can also control interceptors and decide when and which countermeasures to use.

The simulation can handle up to 60 AAW ships. The ships are not moved during the simulation because of their low speeds, compared with aircraft and missiles, and the short duration of a typical AAW encounter.

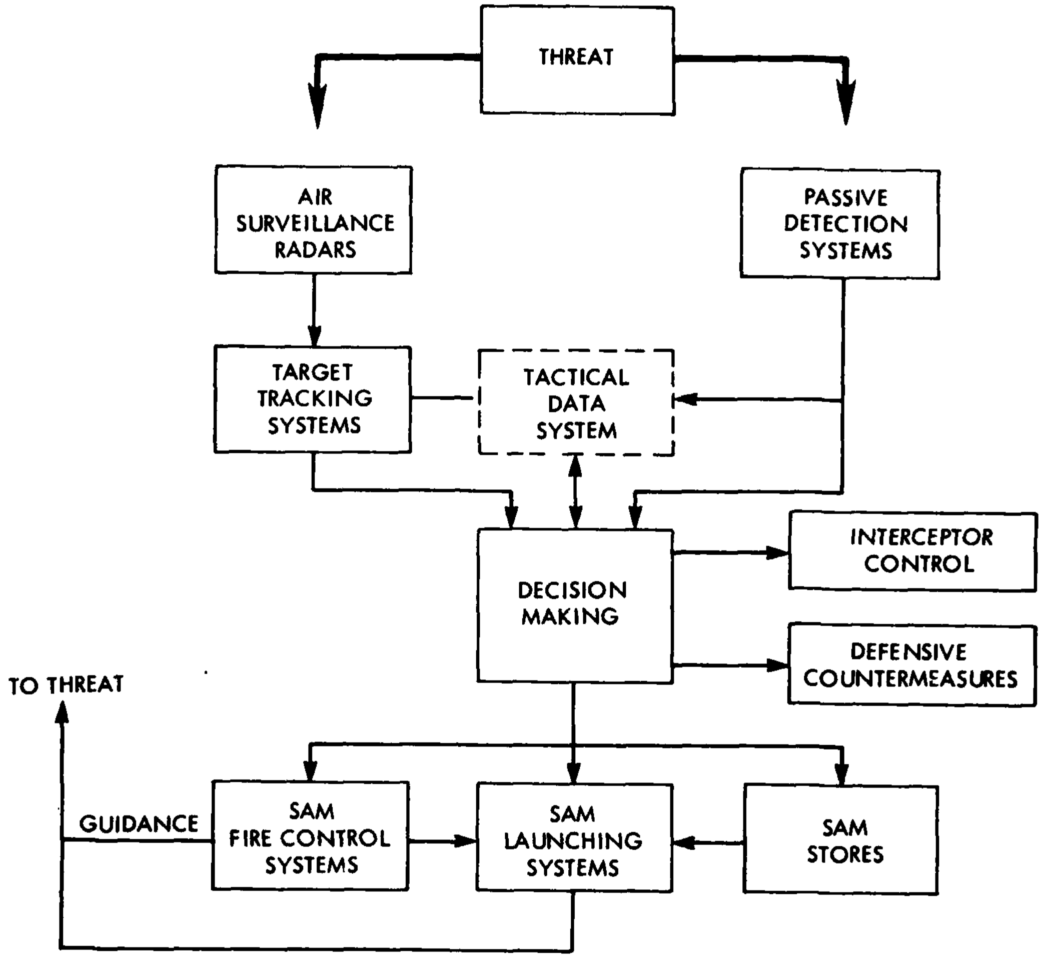


Figure 1-2. Simplified Flow of Ship Functions

Airborne Early Warning

NADS provides for VAW aircraft on airborne early warning stations. AEW aircraft increase the detection range of the force by means of radar and passive detection, by being well forward of the main force and at altitude. The AEW model contains the following components:

Surveillance radar

Automatic and manual detect and track

Air controller function

Interceptor vectoring

Tactical data system

Communications

The level of detail to which each component is modelled varies in proportion to its relative contribution to the total AEW performance. Various VAW aircraft types are accommodated by using different numeric values to characterize these components.

The major AEW functions are shown in Figure 1-3. After detection, the targets are tracked and the air controller function controls the assigned interceptors in their attacks on the targeted Red aircraft. The AEW units receive data from and transmit data to the tactical data net, and they serve as communications nodes between the command center and the interceptors.

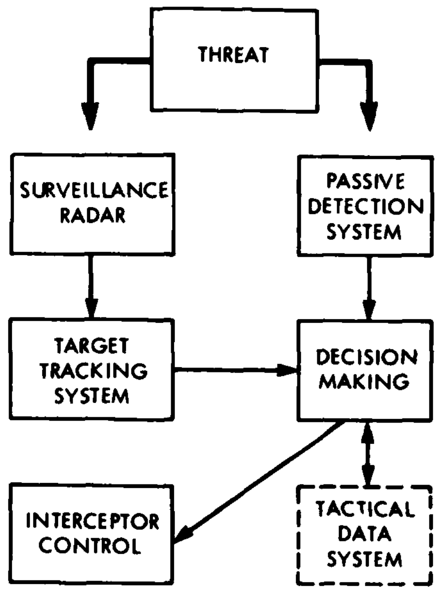


Figure 1-3. Simplified Flow of Airborne Early Warning Functions

The simulation can handle up to six AEW stations. The motion of the AEW aircraft is not modelled, because they are on fixed stations and their positions would change little during an encounter. Fuel consumption is modelled implicitly by keeping track of time on station.

Interceptors

Interceptors are fighter aircraft (VF) capable of detecting and destroying airborne threats with air launched intercept missiles (AIM). Interceptors may be airborne on combat air patrol (CAP) stations or deck launched (DLI). The interceptor module contains the following components:

- Fire control radar
- Target detection and tracking capability
- Decision making
- AIM load
- Self vectoring capability

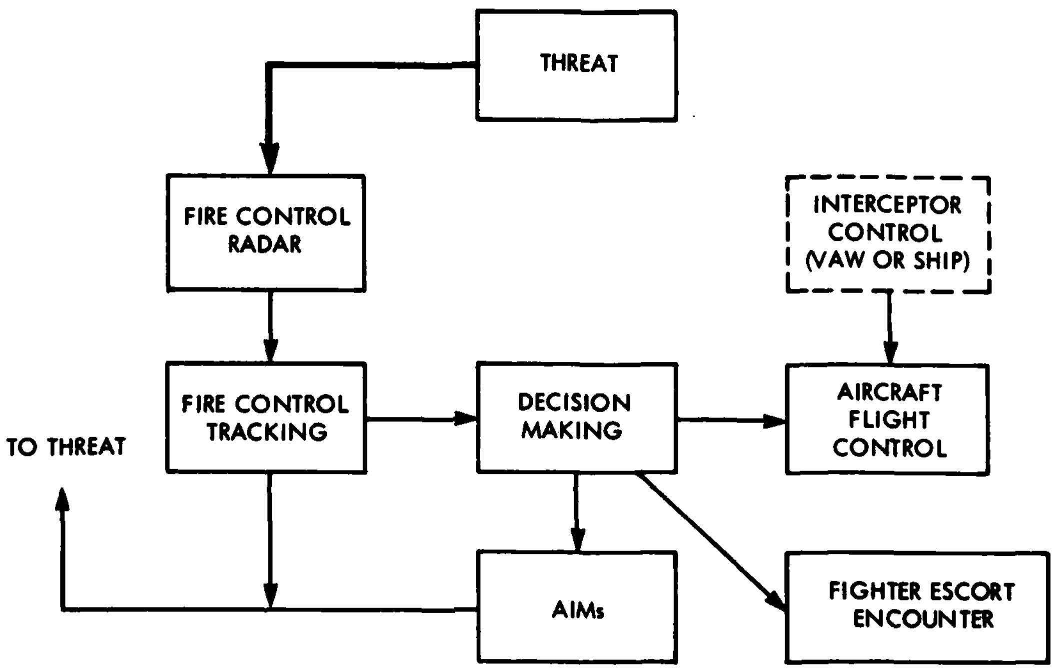


Figure 1-4. Simplified Flow of Interceptor Functions

The level of detail to which each component is modelled varies with the component's relative influence on the total interceptor performance. The various interceptor types are accommodated by using different numeric values to characterize these components.

The major interceptor functions are shown in Figure 1-4. The interceptors are controlled from either ships or airborne controllers. Interceptors can also detect targets and carry out the final part of the attack on their own. The interceptor's decision logic must select among the AIMs available and decide when to launch them. An encounter with a Red fighter escort may occur if escorts are included in the Red flight plan. Time delays, weapon and fuel expenditure, and the outcome of such encounters are modelled.

The interceptor model uses four aircraft speeds - loiter, cruise, normal intercept, and high speed intercept. Appropriate fuel consumption rates are used for each speed. When on CAP station the interceptor motion is not modelled, but fuel is burned at the loiter speed rate. During intercepts the aircraft movement is modelled, and also when flying to CAP stations.

Other Aircraft

Other aircraft may be simulated, in future versions of NADS, to support the AAW mission in roles such as tanker service and electronic warfare aircraft to provide passive detection and defensive jamming capabilities. The principal effects of these aircraft will be modeled, but in less detail than the AEW aircraft and the interceptors.

External Surveillance

External surveillance as used here means any information on the size, composition, route, and timing of an attacking force gained from sources outside the CVBG itself. These sources, which may be satellites, other ships, or land-based sensors, are not individually modelled. The external surveillance is modelled implicitly through introduction of user-defined scenario related surveillance cues. External surveillance cues can be used to influence the decision making, such as surging of VF/VAW aircraft, selection of CAP and AEW station positions, and ship sector assignments.

1.3 RED AIRCRAFT

The behavior of the threat aircraft is directed primarily by user-specified flight plans. Each waypoint of the flight plan is specified by coordinates at the beginning of each leg, and speed to the next waypoint.

Bombers

Bombers follow the preset flight plan through most of the flight. The actual time of launching the ASM can be a function of game conditions, such as defensive jamming. After launching the missiles a simple escape course is set, after allowing for appropriate delays if post-launch guidance is required by the particular ASM type that was launched.

Bombers can be assigned to carry up to four ASMs. The missile type and the target must be specified for each ASM in the scenario.

Reconnaissance Aircraft

If the scenario includes reconnaissance aircraft, they will fly predefined flight plans only. They carry no missiles, but may use defensive self-screen jammers. Their presence may consume defense resources.

Stand-Off Jammer Aircraft

SOJ aircraft carry high power electronic transmitting equipment capable of jamming radars and communication channels from fairly distant positions. The jamming schedule and the intended targets of the jamming are part of the flight plan. Like the recon aircraft, the SOJ carries no weapons, and its entire flight plan is prespecified in the Red scenario.

Fighter Escorts

Fighters, at the scenario writer's option, may accompany the other three types of aircraft to provide protection against interceptors. The simulation does not include the details of dogfights between escorts and interceptors. If an interceptor is diverted by a fighter, the model causes the interceptor to expend fuel, weapons, and time, and it includes the possibility that the interceptor will be destroyed.

Use of Radar

The Red aircraft may use radar for reconnaissance or targeting. The detection of these transmissions can be utilized by the Blue force. On and off times can be randomized or controlled by a predefined schedule in the scenario. (The Red radar simulation is not implemented in the early versions of NADS.)

1.4 ANTISHIP MISSILES

The antiship missiles are either air-launched or SSM. Air-launched missiles enter the game at the bomber's coordinates at launch time. Since the launch point is variable in the game, the ASM flight plan is computed by NADS at the time of launch. The SSM flight plans are computed during the initialization of the run, since the Red surface ships and submarines are modelled as fixed points with fixed launch schedules. The model can simulate a variety of flight profiles.

1.5 DEFENSE STRUCTURE

The AAW defense structure is provided by the input scenario. Disposition of ships and early warning aircraft are fixed. An air plan specifies a fighter posture of fixed Combat Air Patrol (CAP) stations and an on-deck-alert schedule. Fighters are also positioned in response to external surveillance reports. Prosecution of contacts by Battle Group sensors have highest priority followed by response to surveillance messages. Maintaining the input fighter posture is automatic when feasible.

The AAW assets are organized into a layered defense. This defense in depth is organized into three zones - forward air defense, SAM area defense, and terminal defense.

Forward Air Defense

The forward air defense is set up at distances from the force center that permit early detection of bombers and enough time to intercept them before they launch their missiles. Although AEW or surface pickets are the primary means of initial detection in the forward air defense area, interceptors may also make initial detections. The interceptors are held on CAP stations until: (1) they are directed to a target by a controller, (2) they detect a target and attack on their own, or (3) they return to the carrier because of low fuel. The command decision logic compares the known target list with available interceptors and makes rational pairings. Assignments are not limited to one-on-one when the targets outnumber the interceptors.

An air controller normally vectors an interceptor from a CAP station until it has achieved its own detection. When there are not enough interceptors airborne, and the timing permits, deck launched interceptors will be launched. Nearest collision intercept vectors are used. The interceptor will normally detect the targt during this process and prepare to complete the attack. To complete the attack the interceptor selects an appropriate weapon and launch strategy, approaches launch range, fires the weapon, assesses the result, and reattacks if required and if possible. Upon completing an attack, the interceptor will return to the carrier if its fuel or ammunition dictate. If not, it takes up a CAP station at its present position and is available for reassignment.

Air controller detections are shared with other platforms via the tactical data system(s). Tracks and responsibility for targets may be passed to other units if the controller's target handling capacity is exceeded. Surface pickets may also carry SAMs, which will be used if threats enter their envelopes.

SAM Area Defense

Threat aircraft or missiles that penetrate the forward air defense next encounter the SAM area defense, which is provided by a screen of guided missile ships around the vital area. The ships use air search radars for initial detection and identification of targets. Target tracks are transferred to fire control radars and their associated fire control computer channels, which determine target present and future positions, and compute launcher and missile prelaunch orders. Launching and handling systems prepare the SAM for firing. Single or salvo launches are selected, and after launching, the SAM is guided to its target. Methods of guidance vary somewhat from system to system and include home-on-jam, home-all-the-way, mid-course guidance, etc. NADS models these systems using delay times for the principal functions, such as fire control and launcher equipment tie-up times and intercept times. The end game outcome is based on probabilistic data for the missile.

Assigned coverage sectors are defined around each ship. A ship may shoot at any target within its own sector. Depending on the current level of coordination control, ships may also fire into neighboring sectors as directed by the command center. Separation of CAP and SAM areas is defined by a user input range.

Terminal Defense

Those threats that penetrate the forward air defense and the SAM area defense are next engaged by the terminal defense systems of the targeted ships. The terminal defenses are modelled as an aggregate of short range missiles, guns, and electronic countermeasures. Separate off-line studies are relied upon to provide statistical definitions of the effectiveness of these systems for various classes of ships. An effective terminal defense is unlikely against nuclear missiles. The terminal defense phase is used in NADS to establish the number of hits by non-nuclear ASMs and to establish the burst positions of nuclear ASMs.

Area Electronic Warfare

Area electronic warfare uses jamming and deceptive countermeasures on an area wide basis. The primary platform is an EW aircraft orbiting over or near the Blue force center and controlled by the command center's decisions. (AREA electronic warfare is not implemented in the early versions of NADS.)

1.6 COMMAND AND CONTROL

A single command center is used in NADS to represent the Carrier Battle Group's OTC and other delegated AAW-related command functions, such as AAW cordination, EW coordination, and SAM area defense coordination. The command center formulates the required decisions from the information available to it, and implements those decisions by issuing command messages to the relevant Blue units.

Available Information

The command center obtains its input data from three sources: (1) the user-furnished scenario, (2) mesages produced by the various Blue units during the simulation run, and (3) hardware characteristics, which are user-furnished either by manual entry for the particular case or recovered from previously established computer data files.

Scenario Data

The scenario specifies the initial conditions under which the simulation will be conducted. The initial conditions are defined by the following items, which represent prior decisions by the CVBG or higher commands:

Nuclear weapons release status

EMCON status

Positions of CAP stations

Positions of AEW stations

Positions of ships

Number of VF available OWALERT

Limits of VF engagement zone

Assignment of SAM sectors to ships

The CVBG command center model will simulate the receipt of any external surveillance information that the scenario may define, at the user's option. If the scenario specifies any such information, then each simulated surveillance message will contain at least the following data:

Time that the command center receives the message

Time of the surveillance observation

Position of the Red raid element observed

In addition, the surveillance messages may be augmented to contain any or all of the following:

Uncertainty in the position datum

Heading of the raid element

Uncertainty in the heading

Altitude of the raid element

Speed of the raid element

Number of aircraft

Types of aircraft

Distance flown since takeoff

Time elapsed since takeoff

Task Force Message Data

Except as precluded by battle damage or communications jamming, all status messages produced by Blue units in the course of the simulation are accessable to the command center. The Blue unit status messages include the following:

a. From all Blue units:

Change in availability status of the unit

Acquisition and loss of each radar track

b. From all VF in flight:

Acceptance of target assignment

Report if intercept is not possible

Intercept result (kill or miss)

c. From air controllers (shipboard or airborne)

Acceptance of control for assigned VF

Failure of air search radar

Loss of tactical data system capability

Loss of controller capability

d. From ships:

Air search radar failures

SAM system failures

Loss of tactical data system capability

Repair of failed radars

ID of targets being engaged

Acceptance of target assignments

Intercept results

Cannot engage assigned target

e. From aircraft carriers:
   aircraft status reports
   acknowledgement of orders
   on-the-way reports

Hardware Characteristics Data

The technical parameters and performance characteristics of the Blue units and their principal subsystems are provided to support the performance computations required by the simulation. The simulated command center has access to all such technical information on the Blue hardware.

The simulated command center is also provided some technical information on the Red aircraft and missiles for use in the decision making algorithm. A distinction is made between the Red data that the model user chooses to represent the actual performance of the Red systems, and the data that represent the Blue intelligence view of the Red systems. The command center utilizes only the perceived and not the "actual" Red data.

For Red bombers, the command center is provided estimates (or data with which to formulate estimates) of: (1) missile launch range, and (2) probable launch time, given a present position.

For Red missiles, the command center is provided estimates (or data with which to formulate them) of: time to force center from a given present position, and (2) time on target, given the present position and the position of the perceived target.

Decisions

Using appropriately selected portions of the body of available data defined in the foregoing paragraphs, the simulated command center makes the following decisions as they become relevant during the course of the simulation:

Designate targets to VF

Launch VF to fill specified CAP station

Launch VF as DLI

Assign VF to controller

Designate SAM targets to ships

In addition to the foregoing decisions, the model provides for evolutionary growth in capability for centralized combat coordination. The amount of detail in the status messages from ships to the command center is expandable by increasing the number of defined messages, or the contents of the messages, or both. The list of status arrays and decision making subroutines associated with the command center is expandable so that the additional status information can be used to develop and test methods for coordinating intership operations to minimize the number of unengageable or surviving targets.

Communications

The quality and timeliness of tactical decisions at the individual unit level and at the force level depend heavily on the performance of the communication links. NADS does not simulate the communication links, per se, but the delay times associated with the transfer of essential information are modelled. The receipt of each of the messages is represented by user-selectable delay times, and the delays are subject to further degradation by communications jamming and by battle damage. The delay times represent the collective influence of several factors - the time to comprehend a new situation and initiate the appropriate message, the transmission time through the communication link, and the time to comprehend the message content and initiate a response.

Sensors

A wide variety of sensors are used on the various platforms. Radars are the primary means of detecting the threat, and detection range is the most critical characteristic that is modelled. Factors that may vary within an encounter and cause the detection range to change greatly are part of the model. These factors include horizon effects, target size, and jamming. Search radars are represented by full 360-degree coverage except where degraded by jamming. A VF on an intercept sweeps out only a sector. Radars can be scheduled to fail during an encounter as part of the input scenario. Repairs can be scheduled only on shipboard systems; airborne systems will remain in the failed state.

1.7 DEFENSIVE WEAPONS

Two principal categories of defensive weapons used are air-to-air missiles and surface-to-air missiles. Air-to-air missiles are launched from interceptors. Surface-to-air missiles (SAM) are launched from ships. An additional category is the point defense weapons of individual ships, comprising short range missiles and guns.

Air-to-Air Missiles

Four types of airborne intercept missiles (AIM) can be available concurrently on any interceptor. The first type is a long-range, nuclear tipped, active or semi-active missile. The second type has long range performance with a maximum of sophistication and capability. This type can be launched in salvos or one at a time and has either active or semiactive radar guidance. The third type is a shorter range AIM to be used as a backup to the primary weapon, and the fourth is a short range weapon, typified by Sidewinder. Hardware details are not modelled; system performance is characterized as a function of target geometry, firing positions, and other factors that may limit or greatly affect performance.

The model provides for mixed loads. The missile count on each aircraft is maintained throughout a game as a factor in decision making. Actual track of the AIM flight is not simulated, but time and position of impact or miss is computed. Determination of a hit or miss is made at the time of launch, and is based on known (or user postulated) effectiveness data for the AIM.

Surface-to-Air Missiles

A variety of SAM types that differ in range capability, guidance, and warhead are modelled. Control delays, illuminator and fire control channel limitations, as well as launching and handling system delays are included in the model. The effects of target size, altitude, and speed are considered through the use of variable missile coverage envelopes. The actual track of the SAM flight is not simulated, but time and position of intercept is computed. Determination of hit or miss is made at that time for particular type of SAM. The missile count for each SAM type in each ship is maintained throughout a game for decision making and for calculating expenditures.

Terminal Defense Systems

Terminal (or point) defense systems consist of short range surface-to-air missiles, guns, and associated weapon handling and fire control systems. They are not individually modelled, but use effectiveness data from other sources to represent the entire system for each particular class of ship.

1.8 WEAPONS EFFECTS

Non-nuclear Weapons

The effects of a non-nuclear defense weapon against an attacking aircraft or missile are scored as a miss or a hit. The effects of non-nuclear attacking (Red) missiles against the ships of the CVBG are scored as misses or hits, and the damage level on each ship is estimated from the number of hits scored during the simulation run.

Nuclear Weapons

The simulation computes the time and position of each nuclear burst. The input data define the warhead characteristics and the hardness thresholds for each attacking unit and for each defending unit and its major subsystems. Separate thresholds can be specified for damage due to each of eight nuclear environments that are modelled, and for several levels of damage. For each nuclear burst, the simulation examines every Red unit and every Blue unit to determine whether it was within the damage envelopes.

The damage to units and subsystems is applied to the defense capabilities for the remainder of the simulation run. The cumulative effects at the end of the run can be printed out as one of the principal results of the engagement.

2. SOFTWARE DESIGN APPROACH

The approach used in designing the software for the NADS Model is described in the following sections. The topics discussed include programming languages, data considerations, general software structure, and event scheduling.

2.1 PROGRAMMING LANGUAGES

NADS uses a combination of two programming languages, GPSS (General Purpose Simulation System) and FORTRAN. GPSS is a widely available and popular computer language used to model complex systems consisting of many inter-related elements. The language is highly macro in character and provides a tool that enables many types of event-stepped models to be built quickly. GPSS is available on IBM 360 and 370 computers, on the Amdahl 470V/6, the PDP-10, the Burroughs 5700 and 6700, the Univac 1108, and the CDC 6000 series, among others. The language can also be used on the time sharing networks of Computer Science Corporation, National CSS, ADP-Cyphernetics, and University Computing Corporation.

A GPSS program consists of a sequence of GPSS statements, called "blocks". The action defined in a block takes place when a "transaction" moves through it. Transactions are GPSS entities that are defined by a model builder to represent one or more elements of the situation being modeled. They usually represent some moving element, such as a customer moving through a checkout line, or a missile entering an air defense system.

In NADS, GPSS is used for transaction generation; control of logical event sequencing by transaction movement through the system; event scheduling; control of the simulation clock; probability distribution functions; and collection of statistical data. Other GPSS features, such as storage and queues, are used in a few instances where these functions parallel NADS needs.

FORTRAN subroutines are used for functions for which GPSS is not well suited. These include reading input data; performing numerical computations; and maintaining large data files containing status data.

2.2 DATA CONSIDERATIONS

A considerable quantity and variety of data is required for the Model. These data fall into two major catgories; scenario data and technical data. Both types of data are maintained as disk files, which can be accessed during a simulation run.

The scenario data define the initial battle conditions and the pre-scheduled events for a simulation run. Included in the scenario are the disposition of the Blue forces; the Red attack plan; the surveillance messages to be received by Blue from sources outside the game; and definition of other "rules of the game" which have been left to the discretion of the model user.

The second category of data, called technical data, consists primarily of the physical characteristics of various types of hardware. These data are considered to be more static than the scenario data. However, even these values will change occasionally when hardware variations are being studied, or when more accurate data are obtained for previously defined hardware. Maintaining the data in a disk file rather than as hard-coded program constants facilitates such changes.

To reduce the magnitude of the data entry process, software is included to support the preparation of the scenario and technical data files. This software reads free-form input data in convenient units and converts to program variables in internal units.

2.3 GENERAL SOFTWARE STRUCTURE

The general structure of the NADS software is presented in Figure 2-1. The preparation of the scenario and technical data files is shown above the dashed line, as taking place offline to the actual simulation. Below the dashed line are shown the GPSS and FORTRAN programs and the associated inputs and outputs that comprise the simulation software.

The simulation processing is controlled by the GPSS main program. It calls the FORTRAN subroutine, INIT, to load the required scenario and technical data from disk and perform conversions. It calls other FORTRAN

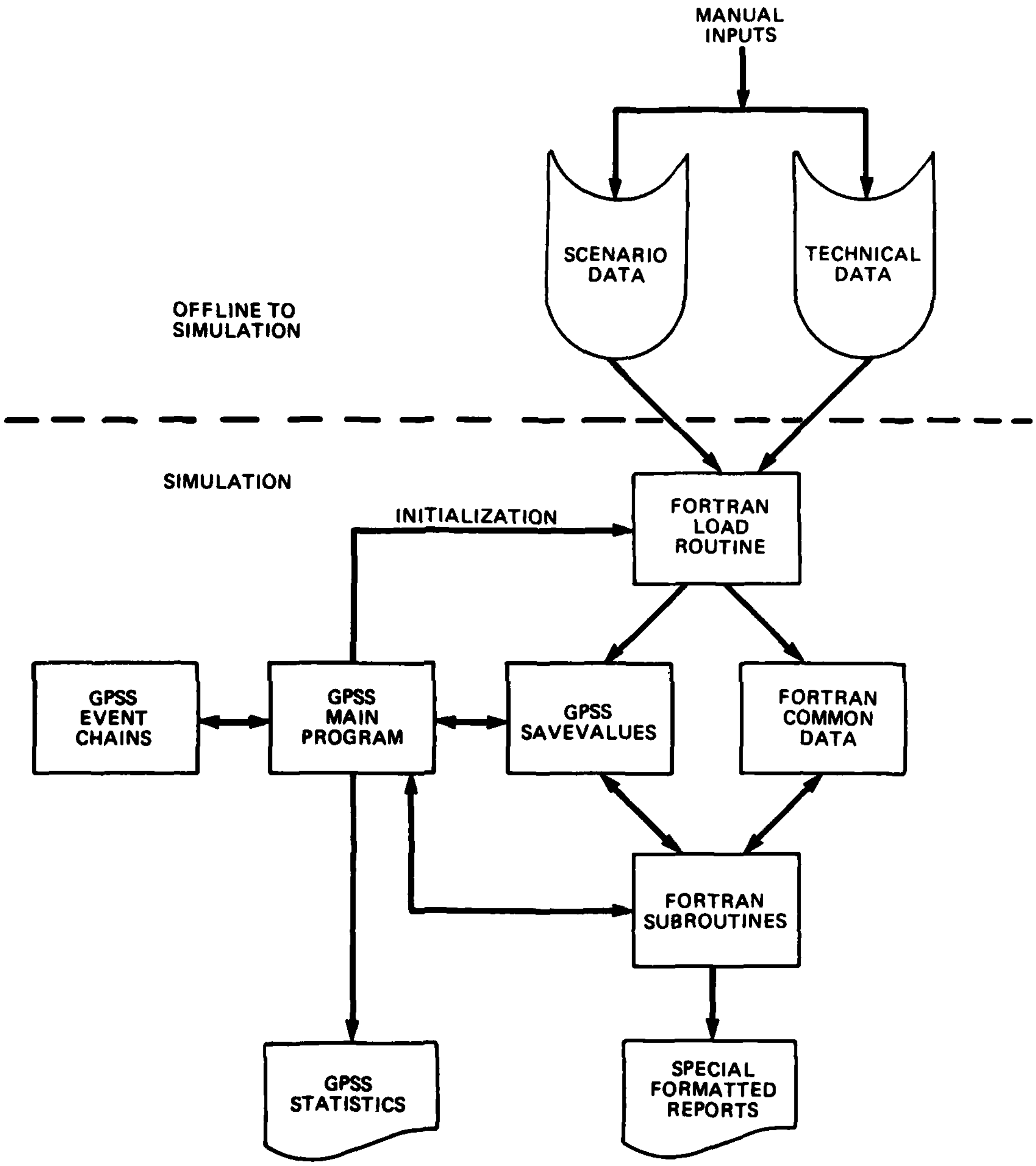


Figure 2-1. General Software Structure

subroutines to perform computations as needed, and to produce any special formatted reports needed from the run. The GPSS program maintains the various event chains that control the sequence and timing of simulation events and accumulates extensive statistics that can be printed at the end of a run.

It should be noted that the scenario and technical input data are loaded into two types of internal data storage, GPSS savevalues and FORTRAN Common. The GPSS program can access only the data defined as GPSS savevalues, which may be individual variables or two-dimensional matrices. The FORTRAN subroutines may access either the savevalues or the FORTRAN Common; however, an address conversion must be performed to enable access to the savevalues. This address conversion is required whenever a GPSS savevalue serves as input or output of a FORTRAN computation and also at startup time when savevalues are initialized with scenario input data (which are read in by the FORTRAN program INIT). The need for address conversion is minimized by keeping as much of the model as feasible in FORTRAN except where GPSS has a clear advantage.

2.4 EVENT SCHEDULING

GPSS makes use of several transaction chains (ordered lists of transactions) in the management of simulation events. The two primary ones are the current events chain (CEC) and the future events chain (FEC). These chains are maintained by the internal GPSS software. Transactions residing on these chains are not directly accessible by a GPSS user program.

In addition, GPSS provides for the creation of one or more user chains by a GPSS program. The program has complete control over its user chains, with the ability to link and unlink transactions as needed. The NADS software makes extensive use of a user chain that is designed to hold tentatively scheduled events. The storage of transactions representing future events on the user chain under program control, rather than relinquishing control to GPSS, enables the program to cancel or reschedule the events based on the occurrence of other prior events.

Tentatively scheduled events are maintained on the user chain, sequenced by event time (and priority level). Transactions are removed from the chain one at a time when the associated event is to occur. Only after one transaction has been processed (moved through as many GPSS blocks as possible at the current clock time), and all other transactions unlinked by this transaction have been processed, is the next user chain transaction unlinked. This provides maximum program access to tentatively scheduled events to enable them to be rescheduled right up to the time of the event.

Table 1-2 summarizes the four principal transaction chains as used in NADS.

Table 2-1. Summary of Principal GPSS Chains as Used in NADS

CHAIN	PURPOSE	SEQUENCE**	CONDITIONS FOR TRANSACTION TO MOVE ONTO CHAIN	CONDITIONS FOR TRANSACTIONS TO Bε REMOVED FROM CHAIN
USER CHAIN 1 (UC1)	User defined. In NADS, used as list of tentatively scheduled events. Transactions can be selectively accessed by a GPSS program to reschedule events based on the occurrence of other events.	User defined. In NADS, ordered by tentative event time* (*Transaction parameter assigned by program).	User controlled by moving the transaction into a GPSS LINK block. In NADS this occurs when the initial or current processing of a transaction is completed and the tentative time of the next event has been computed. The transaction is moved from the CEC to the UC.	User controlled. Transactions are selectively moved from the UC to the CEC by <u>another</u> transaction's movement into a GPSS UNLINK block. In NADS, transactions are removed one at a time from the front of the chain as part of the event step process. Transactions are removed from any part of the chain in order to terminate them or recompute the event
USER CHAIN 2 (UC2)	Temporary storage for sorting & for detection computation at current time.	User defined	User controlled by GPSS link block. In NADS, whenever detection must be recomputed.	User controlled by GPSS unlink block. In NADS, last thing done in current time when event in future is next on UC1.
FUTURE EVENTS CHAIN (FEC)	Interal GPSS chain of definite future events. Not accessible by program.	By priority level within event time* (*As defined by a GPSS ADVANCE or GENERATE block)	GPSS automatically puts the next transaction to be generated by each GENERATE block on the FEC. Transactions are moved from the CEC to the FEC when they move into a positive ADVANCE bllock. The only ADVANCE blocks in NADS are in the CV module and in the DRIVER following the unlinking of the next event.	GPSS automatically moves transactions from the FEC to the GEC when the simulation clock is advanced to the event time for the transaction.

Table 2-1. Summary of Principal GPSS Chains as Used in NADS (Cont.)

CHAIN	PURPOSE	SEQUENCE**	CONDITIONS FOR TRANSACTION TO MOVE ONTO CHAIN	CONDITIONS FOR TRANSACTIONS TO BE REMOVED FROM CHAIN
CURRENT EVENTS CHAIN (CEC)	Internal GPSS chain of transactions to be processed at the current clock time. Not accessible by program.	By priority level	GPSS automatically moves transactions from the FEC to the CEC when the Clock is advanced to the transaction's event time. Transactions are moved from the UC to the CEC when unlinked by another transaction. Copy transactions are put on the CEC when created by the movement of their Parent transaction into a SPLIT block.	GPSS automatically moves transactions from the CEC to the FEC when they move into a positive ADVANCE block. Transactions are moved from the CEC to the UC when they move into a LINK block. Transactions are removed from the CEC (to the GPSS Latent Pool) when they moved into a TERMINATE block.

**In all chains, in case of ties, transactions will be in the order they were linked to the chain within the defined sequence.

3. SIMULATION DESIGN

This section describes the basic design of the simulation in terms of software organization into modules, logical processing flow, and data structure.

Figure 3-1 provides an overview of the NADS software. The software functions are organized into several modules (depicted as blocks in the diagram) which simulate the capabilities and actions of the element that were described in Section 1. These modules are organized into three groups - simulation control, Red attack, and Blue defense - which are described below.

3.1 SIMULATION CONTROL

The Driver Module, which does not correspond to any real world elements, provides program control under GPSS. It generates and processes a Control transaction and a Timer transaction that initializes and closes out, respectively, the processing of a simulation run. The Control transaction also controls the event-stepping process.

As described in Section 2, transactions representing future events are stored on the user chain. Each transaction will have parameters associated with it that define the event time and the address or location of the GPSS code that will simulate the event. The Control transaction unlinks transactions (one at a time) from the front of the user chain when the next event is to occur. The unlinked transaction causes the simulation clock to be advanced to the event time and then transfers to the event address for processing by the appropriate module.

Note that transactions may be linked to the user chain from any part of the model, whenever the next event associated with a transaction is determined and the event time is computed. However, they will always be unlinked by the Control transaction whenever the occurrence of the event is to be simulated unless the event happens before the last transaction UNLINKED.

Transactions are unlinked in other sections of the model only when events are to be rescheduled or cancelled. For example, the detection of

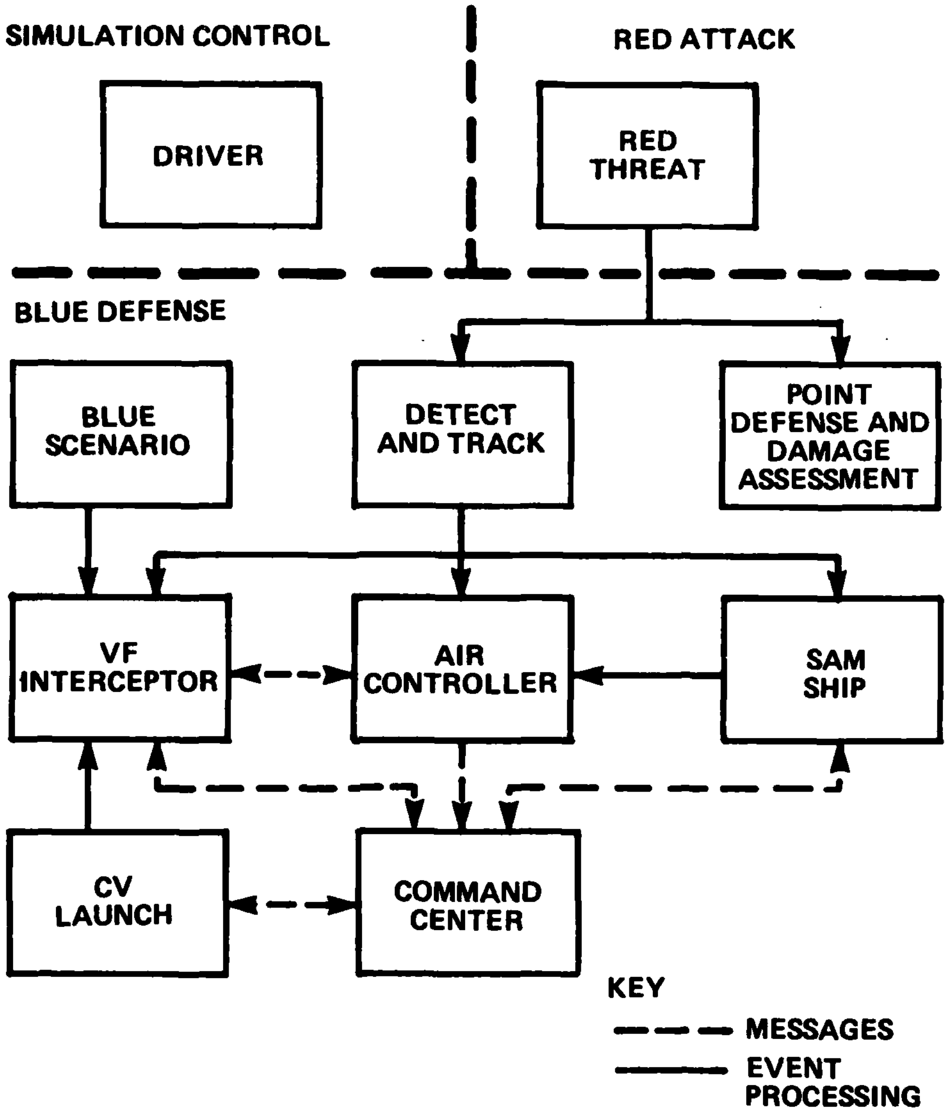


Figure 3-1. Organization of NADS Software Modules

a Red aircraft by a ship may require rescheduling if the Red aircraft changes course. If the Red aircraft is shot down by a VF prior to the detection by the ship, the detection event will be cancelled. To facilitate this type of processing, each transaction on the user chain has parameters assigned that indicate the next scheduled event and provide for the recomputation of event times and for terminating the transaction (thus cancelling all future events for it).

3.2 RED ATTACK

The Red Threat module generates and processes transactions representing the aircraft and missiles comprising the Red attack. One transaction, called a Red parent transaction, will be generated for each Red aircraft and each surface-to-surface missile (SSM) defined in the input scenario. Air-to-surface missiles (ASM) will also be represented by Red parent transactions when they are launched. The term "parent" distinguishes these transactions from the sensor "copy" transactions that are generated for each Blue unit to represent interactions between Red and Blue.

The Red parent transactions are used to simulate events associated with the various nodes of an aircraft or missile flight path. A transaction is terminated if the aircraft or missile is shot down. Otherwise, the last event for an aircraft will be to leave the game (terminate) when it leaves the battle area to return to its base. The last event for a missile will be the warhead burst. When this event occurs, the Damage Assessment module will be executed to determine the effects.

Copies of each Red parent transaction are generated for each Blue unit and sent to the Detect and Track module. These sensor copy transactions are used to simulate the detection of the Red target and the defensive actions taken. Whenever a Red parent event occurs (such as a course change), the events scheduled for that parent's sensor copy transactions (such as detection by a ship) are re-evaluated.

3.3 BLUE DEFENSE

Most of the NADS software is devoted to the simulation of defensive actions taken by the Blue forces. Since the actions of the Blue units will be almost entirely related to the various Red threat elements, these actions are simulated using the sensor copy transaction. As indicated above, one copy of each Red parent transaction is generated for each Blue unit and sent to the Detect and Track module for processing.

The Detect and Track module simulates the detection and tracking functions of all Blue units, based on the detection range and tracking capacity of the individual units.

When the Detect and Track module determines that a Red target is being tracked by a Blue unit, the sensor copy transaction representing that Red threat Blue unit combination is transferred to the appropriate module for decision or action as follows:

Blue Unit Tracking Target	Module Simulating the Decision/Action
VAW	VAW and Air Controller
VF	Interceptor
Ship	SAM Ship or Air Controller

The Command Center module is notified of all tracks, to simulate NTDS functions.

The Air Controller module simulates the control of VF aircraft, for both VAW and ships that have VF under their control.

The SAM Ship module simulates the SAM area defense against all Red targets that get through the forward air defense.

The Terminal Defense and Damage Assessment module simulates the use of defensive countermeasures and short range weapons when Red targts get through the SAM area defense, and assesses the damage caused by conventional and nuclear missiles.

The Interceptor module simulates the decision and actions of VF. While on CAP station, the VF are treated as if they were stationary. However, when on intercept or enroute to a station they become the one type of Blue unit whose movement is simulated. VF may be vectored to a target by an air controller or they may make their own detections and go after their own targets.

The Command Center module simulates the various centralized command functions. The launching of DLI and CAP from carriers is performed in the CV module.

Communications between the various Blue units are simulated via message transactions that are sent between modules to trigger a decision and actions. Delays between the time the message is sent and the time it is received (if at all) are based on the type of communication link involved and any communication jamming* in effect.

The Blue Scenario module initiates other events related to the Blue forces that are independent of the Red threat actions. These include scheduling events for VF that are airborne at game start; surveillance messages that are received from sources outside the game; and equipment failures.* Transactions generated to simulate events of these types are based on the input scenario.

3.4 GPSS TRANSACTIONS

Several types of transactions are used for scheduling and performing the functions described below. Table 3-1 summarizes the transaction types that occur in each module and describes how each type is unused. Table 3-2 defines the contents of the transaction parameters.

*Communications jamming and blue hardware malfunctions are not implemented in NADS version 4.0.

Table 3.1 Transaction Summary

MODULE	TRANSACTION TYPE	PROCESSING
Driver	Control	GPSS and FORTRAN data initialized. Controls event stepping for simulation
	Timer	At the scheduled end of game, initiates the generation of special reports and closes the game.
Red Threat	Red Parents (Aircraft and Missiles)	Controls flight path, radar, jamming, and missile launches of Red aircraft and flight of Red missiles according to input scenario and computations.
		Red missile parent transactions are sent to the Damage Assessment module for processing at the computed time of warhead burst.
	Sensor Copies (Aircraft and Missiles)	Copies of Red Parent transactions are generated for each Blue sensor and sent to the Detect and Track module. These transactions are used to represent tne knowledge and defensive actions of each Blue unit against inaividual Red units.
	Messages (Pseudo)	Pseudo messages are sent to the Command Center when Red units are killed or leave the game and when Red course changes occur. These simulate loss of or changes to target tracks.
Blue Scenario	VF Control	Transactions are generated at the beginning of a game for each VF. For VF enroute to a CAP Station, the transaction is used to schedule its arrival on station. For VF on station, the transaction is used to schedule its return to the CV when its fuel is low. VF control transactions are used to schedule CV events.
	External Surveillance	This message i sent to the Command Center module at the time the next exogenous enemy reort is to be received. The report is read from the input file.
	Command Center Control	Transaction is used to stimulate Command Center decision either at the end of current time or at some future time determined by the Comand subroutine.

Table 3.1 Transaction Summary (Continued)

MODULE	TRANSACTION TYPE	PROCESSING
	Subsystem Failures and Repairs	Transactions representing failure and repair events are sent to the appropriate module (Air Controller, Interceptor or SAM Ship) for processing at the time the event is to occur. (Only ship repairs can occur during the game.)
Detect & Track	Sensor Copies (Aircraft and Missiles)	Determines the time at which a Red threat will be detected by a sensor and when detection will be lost. controls the tracking of detected targets. When tracking of a target is initiated, the corresponding Sensor Copy is sent to the module corresponding to the Blue unit making tne detection (Air Controller, Interceptor or Ship).
Command	Messages (Reports)	Messages are sent to the Command Center whenever tracking is initiated for a new target y any Blue unit, or when contact with the target is lost.
Command Center Module	Messages (Orders and Reports)	Messages will be received from the Detect and Track Module regarding new Red targets being tracked (represents messages from individual Blue units). Messages exchanged with Air Controllers regarding interceptor assignments.
		Messages exchanged with SAM ships rgarding SAM target assignments. Messages exxchanged with CV regarding launching of DLI and CAP.
		Messages received from all Blue units regarding operational status.
	External Surveillance Messages	Messages regarding enemy sightings are used to update data arrays representing knowledge of enemy positions. Depending on the situation, decisions to launch additional VF (or position airborne fighters) may be made.

Table 3.1 Transaction Summary (Continued)

MODULE	TRANSACTION TYPE	PROCESSING
Air Controller (VAW or Ship)	Sensor Copies (Air-craft & Missiles)	Assigns and vectors VF to intercept targets being tracked.
	Messages (Orders and Reports)	Messages are sent between the Controller and the VF regarding fuel, assignments, confirmation, vectoring, and air battles. Messages are sent between the Controller and the Command Center regarding intercept or target assignments and changes in which VF's are under controller.
Interceptor Module (VF)	Sensor Copies (Air-craft and Missiles)	A VF may detect and track his own target or may initially be vectored to the target by an air controller. AIM's are launched when the VF is within the target's LAR. Encounters with bomber escorts may be simulated.
	VF Control	Used to schedule the next event for a VF that is not on intercept. The event will be either arrival on CAP station or return to CV (if on station). These transactions are terminated when a VF goes on an intercept and regenerated (split) when an intercept is complete and he returns to CAP status.
	Messages (Orders and Reports)	Messages are sent between the Air Controller and VF regarding fuel, assignments, confirmation, vectoring, and air battles.

Table 3.1 Transaction Summary (Continued)

MODULE	TRANSACTION TYPE	PROCESSING
SAM Ship Module	Sensor Copies (Aircraft and Missiles)	Ship defense is carried out when a Red missile or aircraft gets through the Forward Air Defense and gets within SAM range. Ships fire on targets in their sectors or as directed by Command Center.
	Messages (Orders and Reports)	Messages are sent between the Ship and the Command Center regarding target assignments and operational status.
Terminal Defense & Damage Assessment Module	Sensor Copies (Aircraft and Missiles)	Red targets that get through the SAM area defense are attacked by short range weapons. Defensive counter-measures will be used.
	Red Parents (Missiles)	Assess damage caused by conventional and nuclear warheads. Terminates Red Parent Missile transactions.
	Messages (Reports)	Messages are split off the Red Parent transaction for each Blue Unit damaged by the missile and sent to the Command Center to report the damage incurred.

Table 3-2. GPSS Transaction Parameters

HALFWORD PARAMETERS:

PH1 - Red Target ID (Aircraft Control) - Aircraft Type with Red ID
PH2 - Time of NEDT Event
PH3 - Event Address
PH4 - Event or Message Type
PH5 - (Sensor) -Time of Lost Detection
- (Red Parent) - Time of dive if point defense occurs first and is scheduled as next event (Aircraft Control) - Station
PH6 - (Sensor) - Time of 2nd Detection (If any due to jamming) (Set negative when a 2nd detection is moved to PH2 as the next event. DETECT must be called again to compute a possible 3rd detection).
-(Red Parent) - Time of Detonation event if scheduled and if next event is dive.
PH7 - (Sensor) - Time of 2nd Lost Detection (if any)
- (Message) - Looping or Control Parameter
PH8 - (Sensor) - Latest time to wait for SAM FCC or LC

BYTE PARAMETERS:

PB1 - Blue Unit ID (Primary) (Making detection, subject of status message, having failure, etc.)
PB2 - Transaction type

1 = Red parent	30 = Control
2 = Sensor Copy	31 = Timer
20 = Message	32 = Blue Failure
21 = External Surveillance Message	33 = VF Control
	34 = SAM Reload
	35 = Blue Damage
	36 = Red Damage
	37 = Command Center Control

PB3 - (Sensor) - Current State Detection/Tracking

0 = Undetected
1 = Detected - in Tracking Queue
2 = Tracking - Self-Generated
2 = Tracking - Passed
(Message) - Blue ID Sending Message

Table 3-2. GPSS Transaction Parameters (Continued)

PB4 - (Message) - Blue ID Receiving Message
- (Sensor) - VF Being Vectored by Controller in PB1

PB5 - (Sensor) - SAM Status (SMSTAT)

0 = Not Processed
1 = Target Engageable
2 = In Queue for FCC
3 = Have FCC - Locking On
4 = In Queue for Launcher Channel
5 = 1 Launcher Channel - In Slew
6 = 2 Launcher Channels - In Slew
7 = Firing, Past Point of No Return (1 LC)
8 = Firing, Past Point of No Return (2 LC's)
9 = Firing, Vertical Launcher
20 = Not Engageable, or Cancelled by CC
21 = Not in Sector
22 = Engaged by Another Ship
23 = Waiting for Response to Self Assign NUC Request

(Message) - Auxiliary Message Data
(Counts, CAP Station No. for New Launch, External Surv. Message No. - Future)

PB6 - (Sensor) - Target Priority (PRIOTY)

0 = Not a Target for this BUID
1 = Threat to Own Ship
2 = CC Assignment
3 = Self Selected, Unconfirmed

3.4.1 Message Transactions

Message transactions use parameter PH4 to define the message content. The code carried on PH4 is defined by the following tables. The messages are categorized by their principal subject matter, with the hundreds digit of PH4 indicating the category:

001 to 099	General Purpose (multiple platform & weapon types)
101 to 199	Aircraft Employment
201 to 299	SAM Employment
301 to 399	Blue Unit and System Status

The message list is given first in Table 3-3 as brief mnemonic labels for convenience in diagrams and commented listings, and then in Table 3-4 with specific definitions.

Table 3-5 displays the transaction parameters for each message type.

Table 3-3. Message Types (PH4 Code and Mnemonic)

General Purpose

	001	Ext Surv
*	002	Detection
	003	Lost Detection
	004	Tracking
*	005	Track Assign
	006	Target Accepted
	007	Target Rejected
	008	Hit Target
	009	Missed Target
	010	Red Change

SAM Employment

201	Target Assign
202	Assign Nuc
203	Self Assign
204	Self Assign Nuc
205	Self Assign NoGo
206	Self Assign Notice

Aircraft Employment

101	Target Assign
102	New Vector
103	Self Assign
104	Self Assign OK
105	Self Assign NoGo
106	On Self Control
107	Red Fighter
108	Intercept Handover
109	Accept Handover
110	Immediate Launch Order
111	VF Engaging Tgt
112	Reject Handover
113	Can't Comply
114	Cancel Launch Order
115	Revise Launch Order
*116	Salvo Fighter
117	Deferred Launch Order

Blue Status

301	Assign Controller
302	Accept Control
303	Unit Down
304	On CAP
*305	TDS Down
306	Controller status change
307	Reject Control
*308	Voice Jam
309	Radar Change
*310	FC Tracker Change
311	FC Channel Change
*312	Launcher Change
313	SAMs
314	Nuc SAMs
315	SAM Local
316	Enroute to Station
317	Go On Cap
318	CV Aircraft Status Report
319	Enroute to CAP

* Not currently implemented.

Table 3-4. Message Definitions

001	This transaction represents the external surveillance message whose serial number is carried in PB5. Read the message from the Blue Scenario using PB5 as the index.
002	Blue Unit, PB1, has radar contact on Red Unit, PH1.
003	Blue Unit, PB1, has lost radar contact with Red Unit, PH1
004	Blue Unit, PB1, is tracking Red Unit, PH1.
005	Blue Unit, PB1, is assigned to maintain track on Red Unit, PH1, and provide principal input to the Tactical Data Net.
006	Blue Unit, PB1, acknowledges its assignment to intercept Red Unit, PH1, and will attempt to do so.
007	Blue Unit, PB1, is unable to intercept Red Unit, PH1, and therefore rejects assignment to it.
008	Blue Unit, PB1, has fired on Red Unit, PH1, and hit it.
009	Blue Unit, PB1, has attempted to intercept Red Unit, PH1, but failed to hit it. Thus, PH1 is a leaker.
010	Red Unit, PH1, flight plan change. Message not jammable.
101	Blue Unit, PB1 (a VF) is assigned to intercept Red Unit, PH1, (at course and speed provided in FORTRAN Common) and attack it with AIMs.
102	Blue Unit, PB1, get new vector from FORTRAN Common for intercept of current target (PH1).
103	Blue Unit, PB1, intends to assign itself to engage Red Unit, PH1, with AIMs.
104	Blue Unit's, PB1, self assignment to Red Unit, PH1, is approved.
105	Blue Unit's, PB1, self assignment to Red Unit, PH1, is overruled.

Table 3-4. Message Definitions (Continued)

106	Blue Unit, PB1, is changing state from controlled intercept of Red Unit, PH1, to self-vectored intercept of PH1.
107	Blue Unit, PB1, is reporting that Red Unit, PH1, is a fighter.
108	Blue Unit, PB4, is assigned as controller of Blue Unit, PB1, (a VF), and PB1's previously assigned intercept of Red Unit, PH1.
109	Blue Unit, PB3, accepts control of Blue Unit, PB1, and of its previously assigned intercept of Red Unit, PH1.
110	VF is to be launched immediately from CV. Red Unit assignment is in PH1. CAP station number can be in PB5.
111	Blue Unit, PB1 (a VF), is engaging Red Unit, PH1.
112	Blue Unit, PB3, rejects control of Blue Unit, PB1, and of its previously assigned intercept of Red Unit, PH1.
113	Blue Unit, PB1 (a CV), cannot comply with launch order.
114	Command Center order to Blue Unit (a CV) to cancel launch.
115	Command Center order to Blue Unit (a CV) to revise a deferred launch time.
116	Not currently implemented.
117	Command Center order to Blue Unit (a CV) to schedule the launch of an aircraft.
201	Blue Unit, PB1, is assigned to engage Red Unit, PH1, with non-nuclear SAM.
202	Blue Unit, PB1, is assigned to engage Red Unit, PH1, with nuclear SAM.
203	Blue Unit, PB1, is assigning itself to engage Red Unit, PH1, with non-nuclear SAM.
204	Blue Unit, PB1, is assigning itself to engage Red Unit, PH1, with nuclear SAM.
205	Blue Unit's, PB1, self assignment to Red Unit, PH1, is overruled.
206	Blue Unit, PB1, will engage Red Unit, PH1, which is threat to own ship.
301	Blue Unit, PB4, is assigned as controller of Blue Unit, PB1 (a VF).

Table 3-4. Message Definitions (Continued)

302	Blue Unit, PB3, accepts control of Blue Unit, PB1 (a VF).
303	Blue Unit, PB1, is Down, i.e., it cannot perform any useful part of its normal mission.
304	Blue Unit, PB1 (a VF), is changing state to "On CAP Station". PH1 = Red Unit if was assigned.
305	Blue Unit's, PB1, Tactical Data System capability is Down.
306	Blue Unit's, PB1, Air Controller vector capability is changed. 0=Down 1=Saturated 2=can accept more
307	Blue Unit, PB3, rejects control of Blue Unit, PB1 (a VF).
308	Blue Unit's, PB1, Voice Link reception is jammed, and some messages or all messages to PB1 may be lost.
309	The Number of air search radars that Blue Unit, PB1, has operating is changed to the number in PB5.
311	The number of Fire Control channels operable in Blue Unit, PB1, is changed to the number in PB5.
313	PB5 is the number of useable conventional SAMs remaining in Blue Unit, PB1
314	PB5 is the number of useable nuclear SAMs available in Blue Unit, PB1.
315	Blue Unit, PB1, is ordered to local control of target selection from any Red Unit in its envelope.
316	Blue Unit, PB1, is VF which was launched. PH1=0 if CAP. PH1=Red ID of target if DLI.
317	Blue Unit, PB1 (a VF), is ordered to halt attack and take up CAP station at current location.
318	Report from PB1 (a CV) on the status of on-deck aircraft.
319	Blue Unit, PB1 (an aircraft) is enroute to station.

Table 3-5. Parameters for Message Transactions (PB2=20)

PH4 MESSAGE TYPE	MESSAGE		PB1 BLUE UNIT ID	PH1 RED UNIT ID	PB3 BLUE ID SENDING MSG.	PB4 BLUE ID RECEIVING MSG.	PB5 AUXILIARY MSG. DATA
001	Ext Surv		-	-	-	CC	Msg. Serial No.
002							
003	Lost Detection		Ship, VAW, VF	Target	PB1	CC	-
004	Tracking		" " "	"	"	"	-
005							
006	Target Accepted	(1)	VF	"	"	Air Contrl	-
		(2)	Ship, Air Contrl	"	"	CC	-
007	Target Rejected	(1)	VF	"	"	Air Contrl	0=Self Vector
		(2)	Ship, Air Contrl	"	"	CC	1=AC Vector
008	Target Hit/Left	(1)	VF	"	"	Air Contrl	-
	Game	(2)	Ship, Air Contrl	"	"	CC	-
009	Missed Target	(1)	VF	"	"	Air Contrl	-
		(2)	Ship, Air Contrl	"	"	CC	-
010	Track Change		-	"	0	"	-
101	Target Assign	(1)	VF	"	CC	Air Contrl	-
		(2)	VF	"	Air Contrl	PB1	-
102	New Vector		VF	"	"	"	-

Table 3-5. Parameters for Message Transactions (PB2=20) (Cont.)

PH4 MESSAGE TYPE	MESSAGE	PB1 BLUE UNIT ID	PH1 RED UNIT ID	PB3 BLUE ID SENDING MSG.	PB4 BLUE ID RECEIVING MSG.	PB5 AUXILIARY MSG. DATA
301	Assign Controller	VF	-	CC	Air Contrl	-
302	Accept Control	"	-	Air Contrl	CC	-
303	Unit Down (1)	"	-	PB1	Air Contrl	-
	(2)	Ship, Air Contrl	-	"	CC	-
304	On CAP (1)	VF	Target (if any)	"	Air Contrl	-
	(2)	Air Contrl	" "	"	CC	-
305						
306	Controller Status Change	Air Contrl	-	PB1	CC	0 =Down 1=Saturated 2=Can accept more
307	Reject Control	VF	0	Air Contrl	"	
309	Radar Change	Ship	-	PB1	"	No. Radars Left
310						
311	FC Channel Change	"	-	"	"	No. Channels Left
312						
313	SAMs	"	-	"	"	No. Conv. SAMs
314	Nuc SAMs	"	-	"	"	No. Nuc SAMs
315	SAM Local	"	-	CC	PB1	-
316	Enroute to Station	VF	0 if CAP Target if DLI	CV	CC	Station if CAP 0 if DLI
317	Go on CAP	"	-	CC	VF	-
318*	CV Aircraft Status Report	CV	-	CV	CC	-
319	Enroute to CAP	VF	Target	CC	Air Control	Air Control VF

Footnote:
* PB3 - CV Number; PB5 - Aircraft Type
PH6 - Number in alert state 2; PH5 - Number in alert state 1;
PB6 - Number in alert state 3.

Table 3-5. Parameters for Message Transactions (PB2-20) (Cont.)

PH4 MESSAGE TYPE	MESSAGE	PB1 BLUE UNIT ID	PH1 RED UNIT ID	PB3 BLUE ID SENDING MSG.	PB4 BLUE ID RECEIVING MSG.	PB5 AUXILIARY MSG. DATA
103	Self Assign Request	VF	Target	VF	Air Contrl	-
104	Self Assign OK	"	"	Air Contrl	PB1	-
105	Self Assign NoGo	"	"	"	"	-
106	On Self Control	"	"	VF	Air Contrl	-
107	Red Fighter (1)	-	"	"	"	-
	(2)	-	"	Air Contrl	CC	-
108	Intercept Handover	VF	"	CC	Air Contrl	-
109	Accept Handover	"	"	Air Contrl	CC	1=Saturated Now 2=Can accept more
110	VF Launch Order	-	0 if CAP Target if DLI	CC	CC	0 if DLI CAP Station No. if CAP
111	VF Engaging Target	VF	Target	Air Contrl	CC	-
112	Reject Handover	"	"	Air Contrl	CC	-
113	CANTCO	CV	"	CV	CC	TYPE A/C
114[1]	Cancel Launch	CV	0	CC	CV	Type A/C
115[3]	Revise Launch	CV	0	CC	CV	Type A/C
117[2]	Scheduled Launch	CV	Target	CC	CV	Type A/C
202	Assign Nuc	"	"	"	"	-
203	Self Assign	"	"	PB1	CC	-
204	Self Assign Nuc	"	"	"	"	-
205	Self Assign NoGo	"	"	CC	PB1	-
206	Self Assign Notice	"	"	PB1	CC	-

Footnotes:
1 PH-6 Time Launch Scheduled
2 PB-6 CAP Number PH5 - Time Launch Desired
3 PH-6 Time Launch Currently scheduled
PH-5 Time Launch Desired

3.4.2 Events Scheduled by Transactions

For transaction types other than messages, parameter PH4 is frequently used to identify the type of event scheduled to occur next for the transaction. (PH4 is not used for certain transaction types that are associated with a single event type and in other instances where the event type is not applicable.) The event codes appearing in PH4 are defined in Table 3-6.

Table 3-7 identifies all events that are scheduled via transactions on the user chain. The events are organized by the module that schedules them. The transaction types used for each event and the contents of the principal parameters are defined, including the GPSS block label/address where the event will be processed.

Events in the 500 series are unique to aircraft handling on the aircraft carriers.

Table 3-6. Transaction Event Codes (PH4)

500 - No future event scheduled
501 - Hook up to carrier launcher
502 - Increase alert level
503 - step down from alert condition
504 - aircraft up - returns to operational readiness status
505 - land aircraft on carrier
510 - launch aircraft
511* - Hook up/immediate launch aircraft
888 - Issue updated assignment orders
900 - Tracking/action decision
901 - New detection, Red course change, State 5 (ready for self intercept), sudden loss of detection
902 - No response to self-assign request
903 - Weapon launch
904 - End of dogfight, VF wins
905 - End of dogfight, VF loses
906 - VF on CAP (arrives on station or arrives at expected intercept point without detecting target
907 - VF returns to CV because of low fuel
908 - Blue Unit loses detection on Red target
909*- Termination of Blue unit (killed or no longer able to carry out mission)
910*- No change in previously schedued event
911 - Blue conventional missile (AIM or SAM) hits Red target
912 - Blue missile misses Red target
913 - End of VF slowdown to let Red target move away for use of long range weapon
914*- Red unit that is an active target for this Blue unit was killed by another Blue unit or left game
915*- VF is waiting for a response to a self-assign request for this target
916*- Code returned by VFCALC for Event 914 when VF is already on CAP (awaiting response to self-assign request for this Red Unit)
918 - Nuclear SAM hits Red target
920 - End of SAM firing evaluation (for miss)
921 - End of SAM firing evaluation (for hit)

*These codes do not represent real events but are used to help route transactions through the GPSS program.

Table 3-7. Scheduled Events by Module

Module Scheduling	PB2 Exact Type	PH4 Event Type	Event	PB1 Blue ID	PH1 Red ID	PH3 Event Address
BLUE SCENARIO	33 (VF Control)	906	VF arrives at CAP station (was on way at start of game)	VF	-	VFP50
		907	VF on CAP returns to CV (low fuel)	VF	-	VFP50
	20 (Ext	1	External Surveillance message received by command center	-	CCM10	TBD
	37 (CC Control	880	Command Center Decision	-	CCM10	
RED THREAT	1 (Red Parent)	0	Red unit enters game (ETYP=1)	-	Aircraft, Missile	RUP10
			Red unit change in velocity vector (ETYP=2)	-		
			Red radar on (ETYP=3)	-	Aircraft	
			Red radar off (ETYP=4)	-		
			Red radar jammer on (ETYP=5)	-		
			Red radar jammer off (ETYP=6)	-		
			Red com jammer on (ETYP=7)	-		
			Red com jammer off (ETYP=8)	-		
			Red aircraft reaches launch line but is jammed and can't launch (ETYP=9)	-		
			Red aircraft leaves game (ETYP=10)	-		
			Red aircraft launches ASM (ETYP=11)	-	Bomber	
			Red missile enters point defense region of target ship (ETYP=12)	-	Missile	
			End of dogfight (ETYP=13)	-	Fighter	RUP55

*The transaction transfers to PDP10 for processing.

Table 3-7. Scheduled Events by Module (Continued)

Module Scheduling	PB2 Exact Type	PH4 Event Type	Event	PB1 Blue ID	PH1 Red ID	PH3 Event Address
DETECT & TRACK	2 (Sensor)	0	Potential detection of Red Target	A/C, Ship	Target	DTS20
		0	Target detection. Initiate tracking if not saturated.			DTS30
		900	Decision on new target	VF		VFS10
				Ship		SSS05
		908	Lost detection on Red target. Stop tracking.	A/C, Ship		DTS50
CV	33 (VF Control)	906	Arrival of newly launched VF on CAP station or at expected intercept (if DLI)	VF	0 if CAP Target if DLI	VFF50
POINT DEFENSE & DAMAGE ASSESS-MENT	1 (Red Parent)	11	Red Missile dive on Blue target (conv. WH)	Ship	Missile	PDP20
		12	Red Missile dive on Blue target (Nuclear WH)			
		1	Conventional Hit on Blue Ship			
		2	Nuclear Burst on Blue Ship			
	35 (Blue Damage)	Max Damage Level	Blue unit suffers damage from nuclear burst	A/C, Ship	0	PDD72
	36 (Red Damage)	Max Damage Level	Red unit suffers damage from nuclear burst	0	A/C, Missile	PDD72

Table 3-7. Scheduled Events by Module (Continued)

Module Scheduling	PB2 Exact Type	PH4 Event Type	Event	PB1 Blue ID	PH1 Red ID	PH3 Event Address	PB5 SAM Status
SAM SHIP	2 (Sensor)	901	Initiate target tracking on Fire Control Channel (if not saturated)	Ship	Target	SSS20	1
			Too late to get FCC and still hit target in SAM envelope			SSS82	2
			FCC is locked on target			SSS32	3
			Too late to get launcher channel and still hit target in SAM envelope.			SSS82	4
		903	Launch SAM(s) (2 Launcher channels if PB5=6)			SSS42	5/6
		908	Lost detection on Red target. Stop tracking.			DRD40	20-22
		911	Conventional missile hits target			SSS49	7/8/9
		918	Nuclear SAM hits target				
		920	End of SAM evaluation (for target miss)				
		921	End of SAM evaluation (for target hit)			SSS56	
	34 (SAM Reload)	-	Reload of SAM Launcher(s) complete.	Ship	Target	SSS60	7/8
VF INTER-CEPTOR	2 (Sensor)	902	No reply was received to Msg 103 (Self Assign Request). VF starts after target on own.	VF	Target	VFS05	
		903	Launch AIM at Red target			N/A	
		904	End of dogfight, VF wins				
		905	End of dogfight, VF loses				
		908	Lost detection on Red target, Stop tracking.			DTS50	

Table 3-7. Scheduled Events by Module (Continued)

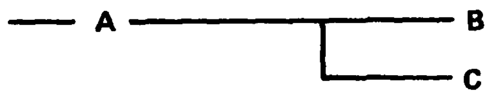
Module Scheduling	PB2 Exact Type	PH4 Event Type	EVENT	PB1 Blue ID	PH1 Red ID	PH3 Event Address
VF INTER-CEPTOR	2 (Sensor)	911	Blue missile hits Red target	VF	Target	VFS05
		912	Blue missile misses Red target			
		913	End of VF slowdown to let Red target move away for long range weapon.			
	33 (VF Control)	906	VF arrives at expected intercept without detecting target	VF	Target	VFF50
		907	VF returns to CV because of low fuel			
		919	Aircraft arrives at altitude	VF	Target	VFF50

3.5 FORTRAN ORGANIZATION

3.5.1 Program Hierarchy

Figure 3-2 presents the program calling heirarchy for NADS. The GPSS program calls HLPRTN whenever FORTRAN computations are required. One of the arguments passed by GPSS is the identifier of the primary FORTRAN routine required.

In the notation used in Figure 3-2,



means that program A calls subroutines B and C.

3.5.2 Programs Versus Common Blocks

Figure 3-3 identifies the common blocks required by each program. This table should be consulted when common block modifications are being considered and to aid in implementing modifications. Table 3-9 is a complete listing of all the FORTRAN common variables.

The user data files are listed below with their associated file numbers. The NADS Users' Manual should be referenced for the format and content of each file.

11 - Blue Sensor Characteristics

12 - Blue Aircraft Characteristics

13 - Blue Aircraft Missile Characteristics

14 - Blue Ship Characteristics

15 - Blue SAM Characteristics

16 - Red Aircraft Characteristics

17 - Red Missile Characteristics

18 - Nuclear Warhead Characteristics

19 - Miscellaneous

20 - CV VF Launch Characteristics/Status

21 - Blue Units

22 - CAP Stations

23 - Red Aircraft Scenario

24 - RED SSM Scenario

25 - Jammer Characteristics

32 - Blue Aircraft Characteristics - Nuclear Vulnerability

34 - Blue Ship Characteristics - Nuclear Vulnerability

36 - Red Aircraft Characteristics - Nuclear Vulnerability

39 - Intelligence Data for Red Aircraft

40 - External Surveillance Messages

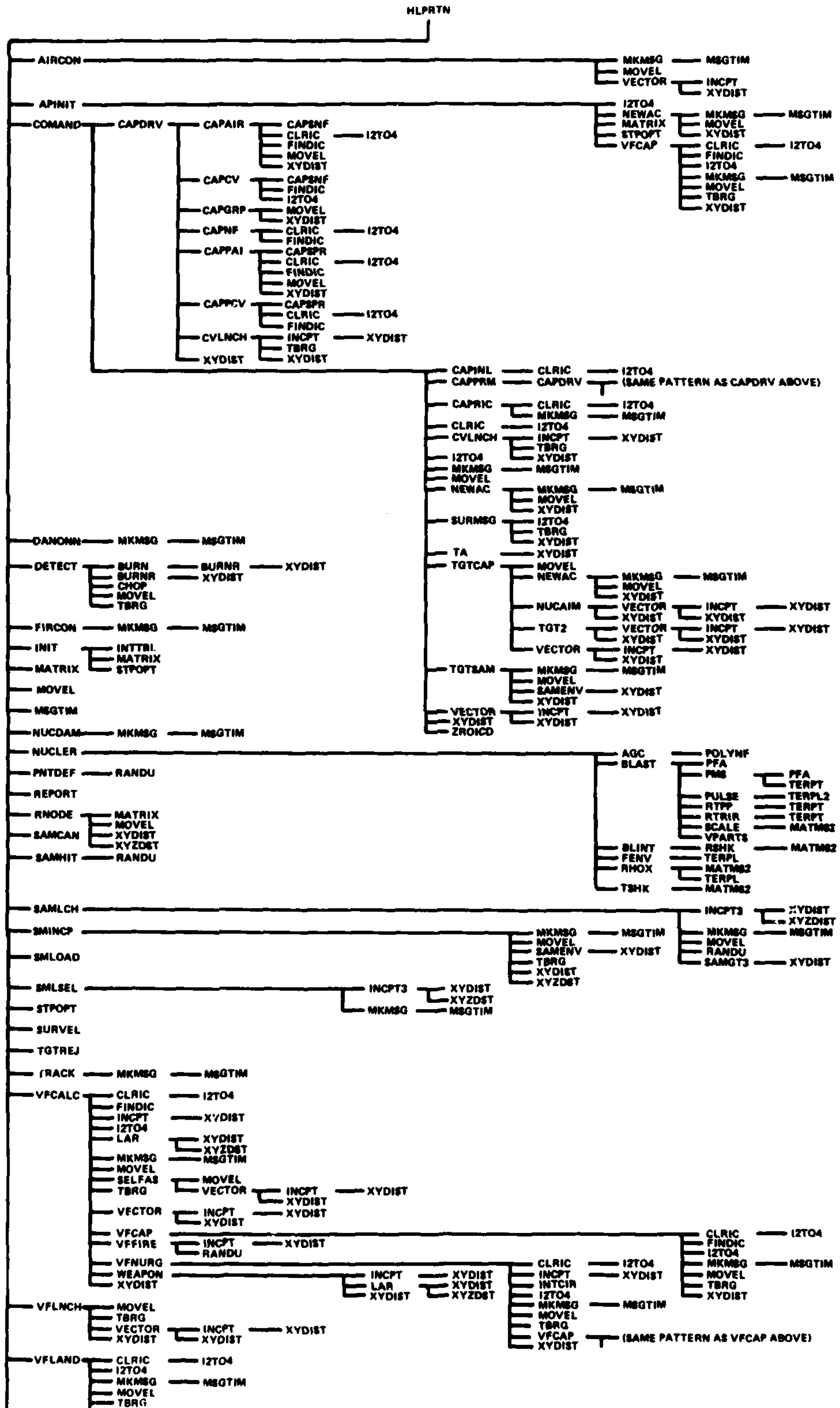


Figure 3-2. Subroutine Calling Structure

Table 3-8. Subprogram Traceback List

SUBPROGRAM	CALLED BY
AGC	NUCLER
AIRCON	HLPRTN
APINIT	HLPRTN
BLAST	NUCLER
BLINT	NUCLER
BURN	DETECT
BURNR	BURN, DETECT
CAPAIR	CAPDRV
CAPCV	CAPDRV
CAPDRV	CAPPRM, COMAND
CAPGRP	CAPDRV
CAPINL	COMAND
CAPNF	CAPDRV
CAPPAI	CAPDRV
CAPPCV	CAPDRV
CAPPRM	COMAND
CAPRIC	COMAND
CAPSNF	CAPAIR, CAPCV
CAPSPR	CAPPAI, CAPPCV
CHOP	DETECT
CLRIC	CAPAIR, CAPCV, CAPINL, CAPNF, CAPPAI, CAPRIC, COMAND, VFCALC, VFCAP, VFLAND, VFNURG
COMAND	HLPRTN
CVLNCH	CAPDRV, COMAND
DANONN	HLPRTN
DETECT	HLPRTN
DLOS	FIRBAL
FENV	NUCLER
FINDIC	CAPAIR, CAPCV, CAPNF, CAPPAI, CAPPCV, VFCALC, VFCAP
FIRBAL	
FIRCON	HLPRTN
INCPT	CVLNCH, VECTOR, VFCALC, VFFIRE, VFNURG, WEAPON
INCPT3	SAMLCH, SMLSEL
INIT	HLPRTN
INTCIR	VFNURG
INTTBL	INIT
I2TO4	APINIT, CAPCV, SURMSG, VFCALC, VFCAP, VFLAND, VFNURG
I4TO2	
LAR	VFCALC, WEAPON
MATM62	RHOX, RSHK, SCALE, TSHK
MATRIX	APINIT, INIT, HLPRTN, RNODE
MKMSG	AIRCON, CAPRIC, COMAND, DANONN, FIRCON, NEWAC, NUCDAM, SAMLCH, SMINCP, SMLSEL, TGTSAM, TRACK, VFCALC, VFCAP, VFLAND, VFNURG
MOVEL	AIRCON, CAPAIR, CAPGRP, CAPPAI, COMAND, DETECT, HLPRTN, RNODE, SAMLCH, SELFAS, SMINCP, TGTCAP, TGTSAM, VFCALC, VFCAP, VFLAND, VFLNCH, VFNURG
MSGTIM	HLPRTN, MKMSG
NEWAC	APINIT, COMAND, TGTCAP

Table 3-8. Subprogram Traceback List (Continued)

SUBPROGRAM	CALLED BY
NUCAIM	HLPRTN, TGTCAP
NUCDAM	HLPRTN
NUCLER	HLPRTN
PFA	BLAST, PMS
PMS	BLAST
PNTDEF	HLPRTN
POLYNF	AGC
PULSE	BLAST
RANDU	PNTDEF, SAMHIT, SAMLCH, VFFIRE
REPORT	HLPRTN
RHOX	NUCLER
RNODE	HLPRTN
RSHK	BLINT
RTPP	BLAST
RTRIR	BLAST
SAMCAN	HLPRTN
SAMENV	SMINCP, TGTSAM
SAMGT3	samlch
SAMHIT	HLPRTN
SAMLCH	HLPRTN
SCALE	BLAST
SELFAS	VFCALC
SMINCP	HLPRTN
SMLOAD	HLPRTN
SMLSEL	HLPRTN
STPOPT	APINIT, INIT, HLPRTN
SURMSG	COMAND
SURVEL	HLPRTN
TA	COMAND
TBRG	CVLNCH, DETECT, SMINCP, SURMSG, VFCALC, VFCAP, VFLAND, VFLNCH, VFNURG
TERPL	FENV, RHOX
TERPL2	PULSE
TERPT	PMS, RTPP, RTRIR
TGTCAP	COMAND
TGTREJ	HLPRTN
TGTSAM	COMAND
TGT2	TGTCAP
TRACK	HLPRTN
TSHK	NUCLER
VECTOR	AIRCON, COMAND, NUCAIM, SELFAS, TGTCAP, TGT2, VFCALC, VFLNCH
VFCALC	HLPRTN
VFCAP	APINIT, VFCALC, VFNURG
VFFIRE	VFCALC
VFLAND	HLPRTN
VFLNCH	HLPRTN
VFNURG	VFCALC
VFTGT	HLPRTN

Table 3-8. Subprogram Traceback List (Continued)

SUBPROGRAM	CALLED BY
VPARTS	BLAST
WEAPON	VFCALC
XYDIST	BURNR, CAPAIR, CAPDRV, CAPGRP, CAPPAI, CVLNCH, COMAND, INCPT, INCPT3, LAR, NEWAC, NUCAIM, RNODE, SAMENV, SAMGT3, SMINCP, SURMSG, TA, TGTSAM, TGT2, VECTOR, VFCALC, VFCAP, VFLAND, VFLNCH, VFNURG, WEAPON
XYZDST	INCPT3, LAR, RNODE, SMINCP
ZROICD	COMAND

Figure 3-3. Programs Versus Common Blocks

NAME OF COMMON / NAME OF ROUTINE	BACHAR	BLCOM	BLUNIT	BMCHAR	CCGRP	CCSTAT	CCTARG	CONST	CPSTAT	CVSTAT	DTCT	ENV	EXTMSG	ICWORK	IMPLIC	INTEL	MESSG	MISC	MVE	NUCLOG	NWCHAR	NWO	PREDCT	RACHAR	RDUNIT	REPRT	RJCHAR	RMCHAR	RSA	SCALT	SHCHAR	SHSTAT	SMCHAR	SNCHAR	TGT	VFSTAT
AIRCON			X			X	X										X		X						X											X
APINIT	X		X			X		X	X	X									X																	X
BLAST	X											X												X				X		X	X					
BLINT												X																								
BURN											X																									
BURNR			X					X			X								X						X									X		X
CAPAIR	X		X			X			X	X				X					X																	X
CAPCV	X								X	X				X				X	X																	X
CAPDRV					X				X					X				X	X																	
CAPGRP	X					X			X	X				X				X	X																	X
CAPINL						X			X					X				X	X																	X
CAPNF	X								X	X				X					X																	
CAPPAI	X					X			X	X				X					X																	X
CAPPCV									X	X				X				X	X																	X
CAPPRM														X					X																	
CAPRIC						X			X					X			X		X																	X
CAPSNF										X				X					X																	
CAPSPR										X				X					X																	
CHOP																																				
CLRIC									X																											X
COMAND	X		X		X	X	X	X	X	X			X	X			X	X	X						X							X			X	X
CVLNCH	X		X					X		X																										
DANONN			X			X											X								X	X					X	X				
DETECT			X					X			X							X	X					X	X		X							X		X
DLOS																																				
FENV												X						X			X	X														
FINDIC									X																											
FIRBAL																				X	X															
FIRCON			X			X											X														X	X				
HLPRTN	X		X			X											X	X	X						X		X									X

Figure 3-3. Programs Versus Common Blocks (Cont.)

NAME OF COMMON / NAME OF ROUTINE	BACHAR	BLCOM	BLUNIT	BMCHAR	CCGRP	CCSTAT	CCTARG	CONST	CPSTAT	CVSTAT	DTCT	ENV	EXTMSG	IUWORK	IMPLIC	INTEL	MESSG	MISC	MVE	NUCLOG	NWCHAR	NWO	PREDCT	RACHAR	RDUNIT	REPRT	RJCHAR	RMCHAR	RSCA	SCALT	SHCHAR	SHSTAT	SMCHAR	SNCHAR	TGT	VFSTAT
INCPT								X																												
INCPT3								X																												
INIT	X		X	X		X		X	X	X			X			X		X	X		X	X		X	X	X	X	X	X		X	X	X	X		X
INTCIR																																				
INTTBL																X																				
LAR				X																					X											X
MATM62																																				
MATRIX																																				
MKMSG																	X																			
MOVEL	X		X																X						X											X
MSGTIM																		X																		
BLKDAT	X	X	X	X	X	X	X	X	X	X	X		X			X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X
NEWAC			X			X											X		X						X											X
NUCAIM	X		X			X		X										X	X						X										X	X
NUCDAM			X			X											X								X			X				X				
NUCLER	X		X	X								X						X		X	X			X	X			X			X		X			X
PFA																																				
PMS																																				
PNTDEF			X															X							X	X		X			X	X				
POLYNF																																				
PULSE												X																								
RANDU																																				
REPORT			X															X		X					X	X						X				X
RHOX																																				
RNODE			X				X	X											X				X	X	X	X	X	X	X		X			X		
RSHK																																				
RTPP																																				
RTRIR	X																																			
SAMCAN		X																													X	X				
SAMENV			X																											X			X			

Figure 3-3. Programs Versus Common Blocks (Cont.)

NAME OF COMMON / NAME OF ROUTINE	BACHAR	BLCOM	BLUNIT	BMCHAR	CCGRP	CCSTAT	CCTARG	CONST	CPSTAT	CVSTAT	DTCT	ENV	EXTMSG	ICWORK	IMPLIC	INTEL	MESSG	MISC	MVE	NUCLOG	NWCHAR	NWO	PREDCT	RACHAR	RDUNIT	REPRT	RJCHAR	RMCHAR	RSA	SCALT	SHCHAR	SHSTAT	SMCHAR	SNCHAR	TGT	VFSTAT
SAMGT3			X															X	X						X						X	X				
SAMHIT																		X							X	X		X								
SAMLCH			X			X											X	X	X						X	X					X	X	X			
SCALE																														X						
SELFAS			X																						X											X
SMINCP			X				X			X							X	X	X						X						X	X	X			
SMLOAD			X															X													X	X				
SMLSEL			X							X							X	X							X						X	X	X			
SURMSG					X								X			X																				
SURVEL								X					X																							
TA					X																				X											
TBRG								X																												
TERPL																																				
TERPL2																																				
TERPT		X																																		
TGTCAP						X	X	X									X	X	X																X	X
TGTREJ			X																													X				
TGTSAM			X			X	X										X	X	X						X						X	X	X		X	
TGT2	X		X			X		X										X	X						X										X	X
TRACK			X			X																														X
TSHK																																				
VECTOR	X		X															X							X											X
VFCALC	X		X					X	X							X	X	X	X						X	X										X
VFCAP	X		X					X	X	X						X		X																		X
VFFIRE	X		X	X														X							X	X										X
VFLAND	X		X			X		X		X								X	X																	X
VFLNCH	X		X			X	X	X	X	X								X	X						X	X										X
VFNURG	X		X					X	X	X						X		X							X											X
VFTGT																																				X
VPARTS												X																								
WEAPON	X		X	X												X		X							X											X
XYDIST																																				
XYZDST																																				
ZROICD														X					X																	

Table 3-9. FORTRAN Common Variables

The FORTRAN common variables are organized into 35 blocks of labelled common. This multiple-page table is arranged with the common block labels in alphabetical order, and each variable in the block is defined.

Most of the FORTRAN common variables have been so named that the first two or three letters of the name indicate which block it is in.

The data types (real, integer, logical) and length (in bytes) of each variable is defined. The righthand column identifies the initialization of each variable that requires it. Constant values are initialized in BLKDAT, the block data program. Where the initial value depends on the input data, the input file number is shown. (These files are named in Section 3.5.2) In some instances the initial value is computed in the program INIT, using data from two or more input files, or in some other program than INIT, as indicated.

To facilitate future updates, insertions, and deletions, the table dedicates a new page to each of the labelled blocks.

Table 3-9. FORTRAN Common Variables (continued)

/BACHAR/

Blue Aircraft Characteristics

N = Blue Aircraft Platform Type, 10 max
J = Nuclear Environment 8 max (see /ENV/)
K = Damage Level 2 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
BACSP(N,4)	Speed (1=Max Endurance, 2=Max Range, 3=Buster, 4=Gate) meters per second	R*4	F-12
BACFBR(N,4)	Fuel Burn Rate at 4 Speeds, kilograms per second		
BACFCR(N,4)	Fuel Consumption Rate, kilograms per meter		
BACFR(N)	Reserve Fuel		
BACFI(N)	Fuel Planned for Intercept		
BACCJT(N)	Comm Jamming Threshold		
BACCTP(N)	Comm Transmitter Power		
BACER(N)	Exchange Ratio (Red Fighters Killed per Blue Fighter Killed)		
BACRSP(N,J,K)	Thresholds of Aircraft Damage Level Response to Nuclear Environments 1 to NENV. A Zero entry for (N,J,1) Makes Environment J Not Applicable to Aircraft Type N. K = 1 is Loss of Mission Capability K = 2 is Loss of Aircraft		F-32
BACLMD(N,2)	Distance Travelled During Climb to Altitude (1=Normal, 2=Minimum Time to Intercept)		F-12
BACLMF(N,2)	Fuel Consumed During Climb to Altitude, for Normal and Fast Climb Profiles		
BACLMA(N,2)	Climbout Altitude for the Two Profiles		
BACNAM(N,2)	Eight-Character Name for Blue Aircraft Type	I*4	
BACSTY(N)	Sensor Type	I*2	
BACADT(N)	Tracking Capacity		
BACTOS(N)	VAW Time On Station Capability		32767
BACFTY(N)	Aircraft Functional Type		F-12
BACLMT(N,2)	Time Consumed During Climb to Altitude, for Normal and Fast Climb Profiles		
BACTDS(N)	Tactical Data System Capability, True or False	L*1	

Table 3-9. FORTRAN Common Variables (continued)

/BLCOM/

(Work Area for Nuclear Scaling)

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
FACT	Scale Factor	R*4	RTRIR
J	Interpolation Index for Function TERPT	I*2	↓
K	Interpolation Index for Function TERPT	↓	↓

Table 3-9. FORTRAN Common Variables (continued)

/BMCHAR/

Blue VF Missile Characteristics, and LAR Data

N = Type Number of Missile, 6 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
BMCRMX(N)	Maximum Range	R*4	F-13
BMCRMN(N)	Minimum Range		
BMCVEL(N)	Average Horizontal Velocity		
BMCPK(N)	Probability of Kill		
BMLTS(N,3)	Three Target Speeds, for LAR Computations		
BMLDZ(N,3)	Delta Z (3 Altitude Differences for LARs)		
BMLAOB(N,3,3,6)	Angle On Bow, 6 Values for Each Combination of Target Speed and Delta Z		
BMLRLA(N,3,3,6)	Radius of LAR, 6 Values for Each Combination of Target Speed and Delta Z		
BMCNAM(N,2)	Eight-Character Name for the Missile Type	I*4	TBD
BMCWH(N)	Warhead Type (0=NonNuc, >0=Index to Nuclear Warhead Characteristics Arrays)	I*2	F-13

Table 3-9. FORTRAN Common Variables (continued)

/BLUNIT/

Blue Unit Status

N = Blue Unit ID (Any Unit Except a VF), 60 max
M = Blue Unit ID (Any Type of Unit), 127 max
K = Red Unit ID, 200 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
BUXC(N)	X-Coordinate	R*4	F-21
BUYC(N)	Y-Coordinate		
BUZC(N)	Z-Coordinate		
BUFTY(M)	Functional Type (1=Ship, 2=VAW, 3=VAQ, 4=VF, Negative=Killed	I*2	0/F-21
BUPTY(M)	Platform Type (Index to Characteristics Arrays)		F-21
BUTNKR	Tanker Fuel Available		F-19
BUPSEN(M)	Air Search Radar Sensor Type		F-12,14
BUNSEN(M)	Number of Sensors in Up Status		INIT
BUDMG(M)	Level of Maximum Damage, Ships 0 to 6, A/C 0 to 2		0
BUNUCJ(M)	Nuclear Environment That Caused Maximum Damage 0=Nonnuc, 1 to NENV for Nuclear		0
BUNUCT(M)	Time That Maximum Damage Occurred		0
ACCTVF(N)	Count of VF Assigned to Air Controller N		0
ACCTVC(N)	Count of VF Being Vectored by Controller N		0
ACMXVF(N)	Max Number of VF That Can Be Assigned to N		F-21
ACMXVC(N)	Max Number of VF That Controller N Can Vector		F-21
BUTRK(M,K)	Track Indicator (True if Blue Unit M is Tracking Red Unit K)	L*1	FALSE
BUTDS(M)	Tactical Data System Capability (T=Up, F=Down)		F-12,14

Table 3-9. FORTRAN Common Variables (continued)

/CCGRP/

Command Center Perceived Status of Red Groups

N = Red Group (or Track) ID Number, 60 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
CGXC(N)	X-Coordinate of the Most Recently Observed Position	R*4	SURMSG
CGYC(N)	Y-Coordinate of the Most Recently Observed Position		
CGXRKO(N)	X-Coordinate of CAP Station That is Positioned in Response to Data on Red Group		
CGYRKO(N)	Y-Coordinate of CAP Station Positioned for Red Group		
CGGID(N)	Red Group Indentifier	I*4	
CGTOBS(N)	Time of Most Recent Observation of Red Group	I*2	
CGTAS(N)	Earliest Time That Red Group is Expected to Arrive at the Associated CAP Station		
CGCNT(N)	Count of Aircraft in the Group,Reported or Assumed		
CGGEN(N)	Generic Airframe Label for Group Members. Index to Intelligence Arrays. (Reported or Assumed)		
CGFT(N)	Functional Type of Red Group Members. (1=Bomber, 2=Fighter, 3=Recce, 4=SOJ)		
CGTOT	Total Number of Red Groups Currently Observed		0
CGMAX	*Maximum Number of Red Groups*		60
CGLCNT(N)	TRUE=Count Observed, FALSE=Count Assumed	L*1	FALSE
CGLGEN(N)	TRUE=Airframe Label Observed, FALSE=Assumed		FALSE
CGLFT(N)	TRUE=Functional Type Observed, FALSE=Bomber Assumed		FALSE
CGLCOV(N)	TRUE=Group Adequately Covered, FALSE=Inadequate		FALSE
CGLNOC(N)	TRUE=No Additional Coverage Required, FALSE=More Coverage Required		FALSE

Table 3-9. FORTRAN Common Variables (continued)

/CCSTAT/

Command Center Perceived Status of Blue Units

N = Blue Unit ID (For Any Unit Except a VF), 60 max
J = VF ID Number 67 max
M = ID of Any Blue Unit (M = J + VFBIAS) 127 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
CCBUID	Blue Unit Containing the Command Center	I*4	F-19
CSNFCC(N)	Number of SAM Fire Control Channels Free	I*2	F-14,21
CSACAV(N)	Air Controller Availability (0=No Capability, 1=Have Capability But Full, 2=Can Control More)		0/F-21
CSACAS(N)	Number of VF Assigned to Controller N		0/NEWAC
CSVFAV(J)	VF Availability (0=NoTargets Assigned, 1=1 Target Assigned, 2=2 Targets Assigned, 3=Out of Game)		0/APINIT
CSACID(J)	Blue Unit ID of Controller Assigned to This VF		0/NEWAC
CSWUTM	Game Time that Next Command Center Decision Will Be Required		32767
CSCACC(J)	Assigned Controller Has Accepted, True or False	L*1	FALSE
CSUNIT(M)	Unit Status (True=Up, False=Down)		F/F-21

Table 3-9. FORTRAN Common Variables (continued)

/CCTARG/

Command Center Perception of Red Target Status

N = Red Unit ID, 200 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
CTCATG(N)	Target Category (-1=Killed, 0=Not Yet Reported, 1=Bomber Before Launch, 2=Bomber After Launch, 3=SSM, 4=ASM, 5=Recce A/C, 6=ECM A/C, 7=Fighter)	I*2	0
CTSASN(N)	Blue ID of Ship Assigned to Target by CC (Set to -2 if Target is Killed or Out of Game)		0
CTNTDS(N)	Number of NTDS Participating Units that are Currently Reporting Contact		0
CTVFID(N)	Blue Unit ID of VF Assigned to Intercept (Set to -1 for a New Launch, Set to -2 if Target is Killed or Out of Game		0

Table 3-9. FORTRAN Common Variables (continued)

/CONST/

(Constants)

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
CONVDG	Degrees to Radians Conversion Factor	R*4	.0174533
CONVFT	Feet to Meters Conversion Factor		.3048
CONVKF	Kilofeet to Meters Conversion Factor		304.8
CONVKN	Knots to Meters per Second Conversion Factor		.51444
CONVLB	Pounds Mass to Kilograms Conversion Factor		.4536
CONVLH	Pounds per Hour to Kilograms per Second Conv. Fact.		.000126
CONVNM	Nautical Miles to Meters Conversion Factor		1852.
RNEVER	Time Later Than End of Game		32767.
TWOPI	Pi * 2		6.2831853
HLFPI	Pi * 1/2		1.5707963
PI	Pi		3.1415927
HRZNK	Horizon Range Factor (Meters Height, Meters Range)		4120.
HZKSQ	HRZNK Squared		16974400.
NEVER	Largest Half-word Value for Setting Half-word Integer to Time Later Than End of Game	I*4	32767
CONVMN	Minutes to Seconds Conversion Factor	I*2	60

Table 3-9. FORTRAN Common Variables (continued)

/CPSTAT/

Blue Aircraft Stations

N = Station Identifier, for Fixed Defense Posture, or for Response to Tactical Situation, 82 Max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
CPXC(N)	X-Coordinate of Station	R*4	0/F-22
CPYC(N)	Y-Coordinate of Station	↓	↓
CPH(N)	Altitude of Station	↓	↓
CPGP(N)	Index of Target Group in Response to Which The Station is To Be Filled (Negative is for Stations of the Fixed Defense Posture)	I*2	↓
CPTIME(N)	Time for which Aircraft Has Been At This Station at Start of Game	↓	↓
CPVFID(N)	Blue Unit ID of Aircraft Assigned to the Station	↓	↓
CPVFTP(N)	Functional Type of Aircraft Required by the Station	↓	↓
CPAS(N)	Alert Status of Aircraft To Fill the Station	↓	↓
CPTOR(N)	Time at which Aircraft Should Respond To Fill the Station	↓	↓
CPTAS(N)	Time at which Aircraft Should Arrive At Station	↓	↓

Table 3-9. FORTRAN Common Variables (continued)

/CVSTAT/

CC Perception of CV Assets

K = CV Index,	5 max	
M = Alert Level,	3 max	
N = Blue Aircraft Type,	10 max	
I = Red Aircraft Airframe Label,	20 max	(1-20)
I = Station Defined for Fixed Defense Posture,	15 max	(21-35)

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
CVBUID(K)	Blue Unit ID for the CV	I*2	0/F-20
CVNAS(K,M,N)	Number of Aircraft of Type N in Alert Level M Available on CV K		0/F-22
CVASTM(K,M,N)	Time Associated with Alert Level M for Aircraft of Type N on CV K		0/F-22
CVNBCV(N,I)	Number of Target Aircraft that Aircraft of Type N Can Cover in an Intercept Mission; Set to 1 for Aircraft of the Types Appropriate for Each of the Fixed Stations		0/F-22,39
CVTOR(K,N)	Time to Schedule Launch for Aircraft Type N On CV K, to Meet a Tactical Requirement		0/CVLNCH
CVTRDX(K*N)	List of Packed Indices for the CV Alert Level, Ordered from Largest to Smallest Time to Schedule Launch		0/CVLNCH
CVCNT	Number of CVs in the Game		0/F-20
CVTORN	Number of CV-Alert Level Pairs in the List of Packed Indices		0/CVLNCH

Table 3-9. FORTRAN Common Variables (continued)

/DTCT/

DETECT Compuations

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
DTXT	X-Coordinate of the Red Unit Relative to Blue Unit	R*4	DETECT
DTYT	Y-Coordinate of the Red Unit Relative to Blue Unit		
DTVXT	X-Velocity of the Red Unit Relative to the Blue Unit		
DTVYT	Y-Velocity of the Red Unit Relative to the Blue Unit		
DTCS2	Radar Cross Section of Red Unit		
DTXRR	Radar Detection Range		
DTDRA	Square of Range Alpha		
DTBID	Blue Unit ID	I*2	
DTVFID	Blue Unit ID - VFBIAS		
DTBFNC	Blue Unit Functional Type		
DTBSNT	Blue Sensor Type		
DTRID	Red Unit ID		
DTJCT	Number of Jammers Turned On Against This Radar		

Table 3-9. FORTRAN Common Variables (continued)

/ENV/

Nuclear Environments

J = 1 , Blast Overpressure
2 , Dynamic Pressure
3 , Overpressure Impulse
4 , Shock Particle Velocity Across Flight Path
5 , Shock Particle Velocity Along Flight Path
6 , Gamma Peak Dose Rate
7 , Neutron Fluence
8 , Thermal Fluence

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
ENV(J)	Intensity of Environment J at the Position of Interest	R*4	
TENV(J)	Game Time that ENV(J) Occurred		
XB	X-Position of Burst		
YB	Y-Position of Burst		
ZB	Z-Position of Burst		
XP	X-Position of Victim at Time of Burst		
YP	Y-Position of Victim at Time of Burst		
ZP	Z-Position of Victim at Time of Burst		
VX	X-Velocity of Victim		
VY	Y-Velocity of Victim		
VZ	Z-Velocity of Victim		
XS	X-Position of Victim at Shock Intercept		
YS	Y-Position of Victim at Shock Intercept		
ZS	Z-Position of Victim at Shock Intercept		

Table 3-9. FORTRAN Common Variables (continued)

/EXTMSG/

External Surveillance Messages

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
EXXC	X-Coordinate of Observed Position of Target Group	R*4	F-40
EXYC	Y-Coordinate of Observed Position of Target Group		
EXZC	Observed Altitude of Target Group		
EXRXU	Standard Deviation of Observed Position Along Semi-Major Axis of Uncertainty Ellipse		
EXRYU	Standard Deviation of Observed Position Along Semi-Minor Axis of Uncertainty Ellipse		
EXRHOU	Orientation of Semi-Major Axis of Uncertainty Ellipse, Relative to Grid North		
EXGID	Target Group Identifier	I*4	
EXTRCV	Time that Command Center is to Receive Message	I*2	32767
EXTOBS	Time that Target Group was Observed		F-40
EXCNT	Number of Aircraft Observed in Target Group (0 = Unknown)		
EXGEN	Airframe Label (Generic Type) Observed in the Target Group (0 = Unknown)		
EXFT	Functional Type Identified in the Target Group (0=Unknown, 1=Bomber, 2=Fighter, 3=Recce, 4=SOJ)		

Table 3-9. FORTRAN Common Variables (continued)

/ICWORK/

Work Area Equivalenced to MVE Common Block

N = Saved Preempt Candidate or Closest Non-feasible Solution, 16 max
M = Number of Aircraft Stations, 82 max
K = Target Group or Station from which VF has been Preempted, 90 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
ICXS	X-Coordinate of Station to be Filled	R*4	
ICYS	Y-Coordinate of Station to be Filled		
ICRNG	Range of the Station from Force Center		
ICTRNF(N)	Time that VF Should Respond to Fill the Station, For Saved Closest Non-feasible Solution	I*2	
ICNTNF(N)	Type of VF To Fill the Station, For Saved Closest Non-feasible Solution		
ICBUNF(N)	CV Number, or Blue Unit ID of Airborne VF, For Saved Closest Non-feasible Solution		
ICCODE(M)	Message Type To Be Issued		
ICAUX(M)	Auxilliary Message Data		
ICPRIG(K)	Index of a Target Group (or of a Fixed Station) From which a VF Has Been Preempted		
ICNTPR(N)	Type of VF To Fill the Station, For the Saved Preempt Candidate		
ICICPR(N)	Index of the Station Position For the Saved Preempt Candiate		
ICTRPR(N)	Time that the VF Should Respond to Fill the Station, For the Saved Preempt Candidate		
ICPRM	Number of Target Groups and Fixed Stations From Which VFs Have Been Preempted		
ICNPRM	Number of Saved Preempt Candidates		
ICNCPR	Number of Target Aircraft Covered by the Saved Preempt Candidates		
ICNNF	Number of Saved Non-feasible Solutions		
ICNCNF	Number of Target Aircraft Covered by the Saved Closest Non-feasible Solutions		
ICIG	Index for Target Group Status Arrays, If Positive; -Index for Fixed Defense Posture Station, If Neg		
ICGEN	1 to 20 = Red Aircraft Airframe Label; 21 to 35 = 20+ Fixed Station Index		
ICTAS	Time that VF Should Fill the Station		
ICTNOW	Current Clock Time		

Table 3-9. FORTRAN Common Variables (continued)

/INTEL/

Intelligence on Red Targets

N = Red Airframe Label, 20 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
INTPSE(N)	Estimated Penetration Speed	R*4	F-39
INTRKO(N)	Keep-Out Range		
INTDST(4,N)	Distance Table for Worst Case Assessments in Absence of Classification Data (1=Bomber, 2=Fighter, 3=Recce, 4=SOJ)		
INTMLR	Estimated Maximum Launch Range for Red Bombers		
INTNE(N)	Number Expected in a Red Group if No Raid Count	I*2	
INTFTS(4,N)	Up to 4 Red Functional Types Applicable to the Indicated Airframe Label		
INTNDX(M,N)	By Functional Type, the Airframe Label to be Assumed for Observed Distances that Equal or Exceed INTDST(M,N)		
INTGEN	Total Number of Red Airframe Labels		0/INIT
INTMXG	Maximum Number of Airframe Labels		20

Table 3-9. FORTRAN Common Variables (continued)

/MESSG/

Messages

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
MSG(15,9)	Message Parameters for up to 15 Messages to be sent. Used by FORTRAN subroutines; HLPRTN transfers parameters to GPSS MSG array prior to returning to GPSS. 1 -- Message Type (PH4) 2 -- Subject of Message (PB1) 3 -- Blue Addressee (PB4) 4 -- Subject Red ID (PH1) 5 -- Auxilliary Message Data (PB5) 6 -- CAP Station Number (PB6) 7 -- Time to Launch Aircraft (PH5) 8 -- Original Action Time (PH6) 9 -- Time of Message Delivery (PH2)	I*2	
NOMSG	Number of Messages in MSG array. Set to zero on Each Call to HLPRTN	↓	0

Table 3-9. FORTRAN Common Variables (continued)

/MISC/

Miscellaneous

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
DDESP	Last Chance Distance for Setting NucAIM Flag	R*4	F-19
DISTT2	"Good Enough" Distance for Secondary Tgt Selection		
MININT	Minimum Permissible Distance from Force Center for VF Intercepts		
NAIMKR	Nuclear AIM Expected Kill Radius		
IPRINT	Print Option for Detailed Simulation Printout	I*4	
JPRINT	Print Option for Initialization Files		
RANDX	Random Number Seed		
RANDY	Next Integer Seed, Computed by RANDU		--
TITLE(20)	80-Character Run Title		F-19
IVERSN(3)	NADS Version Number and Date		4.0 6/14/82
COMTDS	Message Delay Time Between TDS Units	I*2	F-19
COMVCE	Message Delay if Either Unit is Non-TDS		
TWAITC	Time that Command Center Awaits Confirmation of a Target Assignment, before Reassigning it.		
TWAITL	Time that Command Center Awaits Confirmation of a New VF Launch And Target Assignment		
TWAITF	Time a VF Awaits Response to Self-Assign Request		
TMARG	Time Margin for DLI Launch Estimation		
MNDOGF	Mean Duration of a Dogfight		
SDDOGF	Std Deviation of Dogfight Duration		
NUCNO	Number of Targets Close to Primary Target, that will Set NucAIM Selection Flag		
CDOCT	SAM Coordination Level (1=Command Center, 2=Ship Sectors, 3=No Coordination)		
TDUO	Time between 2 Rounds of SAM Salvo		
TMINS	SAM Envelope Time Threshold, For Quick Reaction		
TEVALS	Time to Evaluate SAM Hit or Miss		
TOOLT	"Too Late" Time for Cancel or Reschedule A/C Launch		
NUCREL	Nuclear Weapons Release for Force	L*1	
SALVO2	VF Launch Policy for Tvpe 2 Missiles (T=Two-missile Salvo, F=Single)		
SALVO3	VF Launch Policy for Type 3 Missles		
PRTOPT(16)	Print Option Switch Vector for Detailed Printouts (TRUE = IPRINT > PRTOPT Index)		

Table 3-9. FORTRAN Common Variables (continued)

/MVE/

Move List

N = Number of VFs, 67 max
M = Number of Red Units, 200 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
BLUID(N,2)	List of VF IDs for which Position and Fuel Update is to be performed	I*2	
REDID(M,2)	List of Red IDs for which Position Update is to be performed	↓	

Table 3-9. FORTRAN Common Variables (continued)

/NUCLOG/

Log of Nuclear Bursts

I = Index Number of Burst, 150 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
NUCTIM(I)	Time of Nuclear Burst	R*4	
NUCX(I)	X-Position of Burst, kilometers		
NUCY(I)	Y-Position of Burst		
NUCZ(I)	Z-Position of Burst		
NUCTFB(I)	Time that Fireball Masking Effect Fades Out		
NUCTYP(I)	Warhead Type Number	I*2	
NUCN	Index of Latest Entry (No. of Bursts)		0
NUCI	Index of Earliest Burst whose Fireball Could Still Be Active		1

Table 3-9. FORTRAN Common Variables (continued)

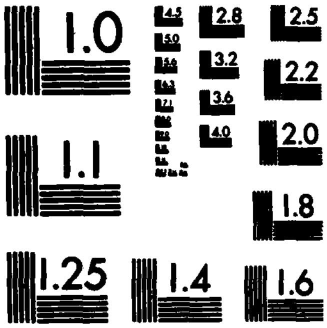
/NWCHAR/

Nuclear Warhead Characteristics

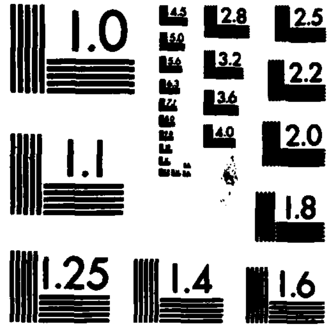
I = Warhead Type Number, 10 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
NWCYLD(I)	Burst Yield, Kilotons	R*4	F-18
XNEUT(I)	Neutron Output, n/KT		
SN(I)	Neutron Source Level, n		INIT
XN14F(I)	Fraction of Neutrons Faster than 10 MeV		F-18
XPGAM(I)	Prompt Gamma Energy Fraction		
GAMDT(I)	Effective Prompt Gamma Pulse Width, Seconds		
GDOSE	Gamma Fluence-to-Dose Conversion Factor		
SG(I)	Effective Gamma Source Level		INIT
THERF	Thermal Fluence Multiplication Factor		F-18
ST(I)	Effective Thermal Source Level		INIT

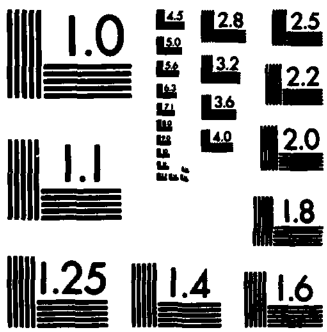
AD-A120 477 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) 2/4
(U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA WATERWHEEL
PROGRAM OFFICE R W COVEY ET AL. 15 SEP 82
UNCLASSIFIED N00014-81-C-0715 F/G 15/3 NL



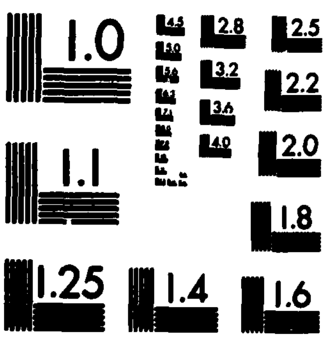
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



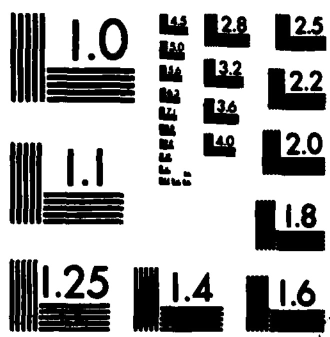
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Table 3-9. FORTRAN Common Variables (continued)

/NWO/

Nuclear Warhead Tables

N = Index for Neutron Table, 40 max
I = Index for Gamma Table, 40 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RTN(N)	Table of Air Mass Integrals, gm/sq cm	R*4	BLKDAT
TN(N)	Neutron Buildup Factors vs RTN		
RTG(I)	Table of Air Mass Integrals, gm/sq cm		
TG(I)	Gamma Attenuation Factors vs RTG		
NON	Number of Data Points Filled in RTN,TN	I*2	
NOG	Number of Data Points Filled in RTG,TG		

Table 3-9. FORTRAN Common Variables (continued)

/PREDCT/

Attack Prediction

N = Red ID for a Bomber or Fighter , 60 max
M = Red ID for Any Red A/C or Missile, 200 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
PRTSTO(N)	Time Since Takeoff	R*4	0.
PRDSTO(N)	Distance Flown Since Takeoff	↓	0.
PRTLLN(N)	Nominal Time to Launch Line	I*2	0
PRTCMR(N)	Latest Time to CAP Max Intercept Range	↓	0
PRTDMR(M)	Latest Time to DLI Max Intercept Range	↓	0
PRTMIN(M)	Time at Minimum Intercept Range from Force Center	↓	0
PRFLAG(M)	Set True whenever Velocity Changes; Reset False when New Prediction is made	L*1	F

Table 3-9. FORTRAN Common Variables (continued)

/RACHAR/

Red Aircraft Characteristics

N = Red Aircraft Platform Type, 20 max
J = Nuclear Environment, 8 max
K = Damage Level, 2 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RACRCS(N)	Radar Cross Section	R*4	F-16
RACRSP(N,J,K)	Thresholds of Aircraft Response to Nuclear Environment J. Zero entry for (N,J,1) means that Environment J is Not Applicable to Type N. K=1 is Loss of Mission Capability K=2 is Loss of Aircraft		F-36
RACMTO(N)	Total Missiles Carried	I*2	F-16
RACMTY(N,4)	Missile Type for up to 4 Missiles, in the Order of Launching		
RACDBL(N)	Delay Between Missile Launches, Seconds		
RACDLL(N)	Delay After Last Launch (for Guidance)		
RACJTY(N,6)	Jammer Type for up to 6 Jammers		

Table 3-9. FORTRAN Common Variables (continued)

/RDUNIT/

Status of Red Units

N = ID of Any Red Unit (Aircraft or Missile), 200 max
M = ID of a Red Aircraft, 60 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RUXC(N)	X-Coordinate	R*4	F-23,24
RUYC(N)	Y-Coordinate		
RUZC(N)	Z-Coordinate		
RUXS(N)	X-Speed		
RUYS(N)	Y-Speed		
RUZS(N)	Z-Speed		
RUJWMH(M,6)	Power Density of Jammer, W/MHz		
RUFTY(N)	Functional Type (Negative=Killed, 0=Not in Game, 1=Bomber, 2=Fighter, 3=Recce, 4=SOJ, 5=SSM, 6=ASM)	I*2	F-23,24
RUPTY(N)	Platform Type or Missile Type		
RUMTY(N)	Equivalent Name for RUPTY Array		
RUTLU(N)	Time of Last Update		
RUNODE(N)	Next Node of Flight Plan		1
RUTARG(N)	Blue ID of Target (FTY = 5 or 6)		0
RUIDL	Last Red ID Number to Enter Game, so far		0
RUTGAP(N)	Time in Gap		
RUNUCT(N)	Time that Maximum Damage Occurred		0
RUDMG(N)	Level of Maximum Damage		0
RUNUCJ(N)	Nuclear Environment that Caused Maximum Damage (Zero is Nonnuclear)		0
RUGID(M)	ID of Red Group with which the Aircraft is Associated (Zero is No Association)		0
RUJAM(M,6)	Blue Sensor Type Targeted by Jammers 1 to 6 (Zero is Jammer OFF)		0
RUACCT	Number of Aircraft in the Red Aircraft Scenario		F-23
RUTDNV(N)	Time of Next Change in Velocity Vector, for Updating Detection Times		

Table 3-9. FORTRAN Common Variables (continued)

/REPRT/

Status Reports Summary

R = Red ID (Aircraft), 60 max
N = Red ID (Any Red Unit), 200 max
B = Blue Unit ID (Any Non-VF Unit), 60 max
V = VFID (= Blue Unit ID - VFBIAS), 67 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RASM(R)	Number of ASMs Launched	I*2	0
RUFTYP(N)	Red Functional Type (1=Bomber, 2=Fighter, 3=Recce, 4=SOJ, 5=SSM, 6=ASM)	↓	
BSAMC(B)	Number of Conventional SAMs Fired	↓	0
BSAMN(B)	Number of Nuclear SAMs Fired	↓	0
VENG(V)	Number of Targets Engaged by VF	↓	0
VAIM(V,4)	Number of AIMs Fired, Types 1 to 4	↓	0
VLNCH(V)	Number of times Launched from CV During Game	↓	0
RKILLV(N)	Blue ID of Killing VF	↓	0
RKILLS(N)	Blue ID of Killing Ship	↓	0
RPTDEF(N)	Blue ID of Ship whose Point Defense was Penetrated	↓	0
RCHIT(N)	Blue ID of Ship Hit by Conventional Missile	↓	0
RNHIT(N)	Blue ID of Ship Targeted by Nuclear Burst	↓	0
RENGVF(N)	True = Engaged by a VF	L*1	False
RENGSH(N)	True = Engaged by a Ship	↓	False
VKILLF(V)	True = Killed in Dogfight	↓	False

Table 3-9. FORTRAN Common Variables (continued)

/RJCHAR/

Red Jammer Characteristics

N = Red Jammer Type, 8 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RJCPOW(N)	Total Effective Radiated Jamming Power, Watts	R*4	F-25
RJCMXW(N)	Maximum Bandwidth, MHz	↓	↓
RJCMNW(N)	Minimum Bandwidth, MHz	↓	↓

Table 3-9. FORTRAN Common Variables (continued)

/RMCHAR/

Red Missile Characteristics

N = Red Missile Type, 10 max
J = Nuclear Environment, 8 max
K = Damage Level, 3 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RMCRMX(N)	Maximum Range	R*4	F-17
RMCRMN(N)	Minimum Range		
RMCACL(N)	Climb Angle		
RMCATD(N)	Terminal Dive Angle		
RMCVCL(N)	Climb Velocity		
RMCVCR(N)	Cruise Velocity		
RMCVTD(N)	Terminal Dive Velocity		
RMCHCR(N)	Cruise Altitude		
RMCRCS(N)	Radar Cross Section		
RMCPSF(N)	Probability of Salvage Fuze Firing, Given a Hit by a Blue Weapon		
RMCRSP(N,J,K)	Thresholds of Vulnerability to Nuclear Environments K = 1, Missile Airframe or Guiddance System is Damaged. Missile Killed if Nonnuclear. K = 2, Salvage Fuze is Fired K = 3, Warhead Disabled Zero for K=1 makes J Non-applicable. 1.E60 for K=2 Precludes Salvage Fuzing		F-37
RMCHOB(N)	Planned Height of Burst. Zero for Nonnuclear.		F-17
RMCPWH(N)	Probaility that Nuclear Warhead or Fuze was Killed, given that the Missile Itself suffered a Hard Kill Without firing Salvage Fuze.		
RMCWH(N)	Warhead Type (Zero is Conventional; Positive is index to NWCHAR Tables)		

Table 3-9. FORTRAN Common Variables (continued)

/RSA/

Red Scenario, Aircraft

N = Red ID (for a Red Aircraft), 60 max
M = Red Aircraft's Missile Number, 4 max
L = Node of Aircraft Flight Path, 20 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
RSAXC(N,L)	X-Coordinate	R*4	F-23
RSAYC(N,L)	Y-Coordinate		
RSAZC(N,L)	Z-Coordinate		
RSAV(N,L)	Velocity		
RSATRG(N,M)	Blue ID of Target of Mth Missile	I*2	
RSANCT(N)	Count of Nodes Specified		
RSANTY(N,L)	Node Type 1 = Enter Game 2 = Change in Velocity Vector 3 = Radar ON 4 = Radar OFF 5 = Radar Jammer ON 6 = Radar Jammer OFF 7 = Communications Jammer ON 8 = Communications Jammer OFF 9 = Bomber Arrives at Launch Line 10= Leaves Game		
RSAJSN(N,6)	Blue Sensor Type Targeted by Jammers 1 to 6 in Node Type 5		

Table 3-9. FORTRAN Common Variables (continued)

/SCALT/

Scaling Factors for 1-KT Sea Level Nuclear Bursts

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
AK	Scaled Altitude	R*4	
HBK	Scaled Height of Burst		
PSCL	Scaled Atmospheric Pressure, in Atmospheres		
RSCL	Scaled Range		
YSCL	Scaled Yield (cube root of KT)		
RHOA	Atmospheric Density, Slugs per cu. ft.		
AO	Function of PO and RHOA		
PO	Atmospheric Pressure, psi		

Table 3-9. FORTRAN Common Variables (continued)

/SHCHAR/

Ship Characteristics

N = Ship Platform Type (Class), 10 max
J = Nuclear Environment, 8 max
K = Damage Level, 6 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
SHCCJT(N)	Communication Jamming Threshold	R*4	---
SHCCTP(N)	Communication Transmitter Power		---
SHCRIL(N)	Transition Range for Illuminator Tie-up Time		F-14
SHCRAC(N)	SAM Action Range		---
SHCHCP(N)	Hot CPA (Implies Threat to Own Ship)		F-14
SHCRSP(N,J,K)	Nuclear Damage Threshold Levels, for Response to Each Nuclear Environment. K=1 is Radar Down, K=2 is 50% Wpn Delivery Impairment, K=3 is 100% Wpn Delivery Impairment, K=4 is 90% Mobility Impairment, K=5 is 90% Seaworthiness Impairment, K=6 is Ship Destroyed (Sunk)		F-34
SHCDPD(N)	Action Radius of Point Defense Hemisphere		F-14
SHCPKP(N,2)	Point Defense System Kill Probability against Sea Skimmer and Diving Type Missiles		
SHCECM(N)	Fraction of Point Defense Kills Attributable to ECM		
SHCMCT(N,2)	Count of SAMS Remaining (1=Conventional, 2=Nuclear)	I*2	
SHCMTY(N,2)	SAM Types (Conv and Nuc)		
SHCSTY(N,4)	Sensor Types of 1 to 4 Search Radars		
SHCADT(N)	Total Tracking Capacity of Search Radars		
SHCNOC(N)	Number of Fire Control Channels Operable		
SHCLTP(N)	SAM Launcher Type (1=Single Rail, 2=Dual Rail, 3=Vertical Launching System		
SHCNOL(N)	Number of SAM Launchers Operable		
SHCTLK(N)	Lock-on Delay for FC Radar after target designation		
SHCILL(N)	Number of SAM Illuminators (or Guidance Channels)		
SHCSLW(N)	Average Launcher Slew Time		
SHCTLD(N)	SAM Launcher Reload Time		
SHCTWT(N)	Estimated Waiting Time for an Illuminator, used to Predict Shoot-Look-Shoot Opportunity		
SHCTIL(N,2)	Iluminator Tie-up Times for Short Range, and for Long Range Intercepts		
SHCHIT(N,2)	Number of Hits Required by Conventional Antiship Missles to cause 50% and 100% Wpn Del Impairment		
SHCTDS(N)	Tactical Data System Capability, T or F	L*1	
SHCCC(N)	Command Center Capability, T or F		

Table 3-9. FORTRAN Common Variables (continued)

/SHSTAT/

Ship Status

N = Blue Unit ID of Ship, 60 max
L = Illuminator or Guidance Channel, 8 max
K = Fire Control Channel, 8 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
SSSECT(N,2)	Ship Self-assignment Sector, CCW and CW limits	R*4	F-21
SSPKPD(N,2)	Point Defense System Pk against Sea-Skimmers, and Diving type missiles		
SSMTOT(N,2)	Total SAMs Remaining in Ship, 1=Conv, 2=Nuc	I*2	F-14/21
SSCHUP(N)	Number of Fire Control Channels in UP status		
SSCHAV(N)	Number of Fire Control Channels Free for Use		
SSFSTA(N,K)	Fire Control Channel Status (1=Available, 2=Occupied Preemptable, 3=Occupied but Not Preemptable, 4=Post Launch Occupied)		1
SSFTRG(N,K)	Red IDs of Targets to which Fire Control Channels are assigned		0
SSNLUP(N)	Number of Launcher Channels (Rails) UP		F-14/21
SSLSTA(N,2)	Launcher State		
SSLTRG(N,2)	Red ID of Target to which Launcher is Assigned		0
SSLLOD(N,2)	Missiles Loaded on Launcher (0=Empty, 1=One Conv, 2=Two Conv, 21=One Nuc, 22=Two Nuc)		
SSGCUP(N)	Number of Guidance Channels in UP Status		F-14/21
SSTGAV(N,L)	Time that Guidance Channel (or Illuminator) Will Be Available		0
SSCHIT(N)	Count of Hits on Ship by Conventional Warheads		0
SSCNT	Total Number of Ships in Game		F-21
SSNUCD(N)	Ship has Local Nuclear Weapons Release (T or F)	L*1	

Table 3-9. FORTRAN Common Variables (continued)

/SMCHAR/

SAM Characteristics

N = SAM Type, 10 max
T = Red Missile Target Type, 11 max
(T=11 is any Red Aircraft)

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
SMCMXR(N)	Maximum Range	R*4	F-15
SMCSPD(N)	Speed (Average Horizontal Component)		
SMCMXC(N,T)	Maximum Cross Range		
SMCMNC(N,T)	Minimum Cross Range		
SMCMNR(N,T)	Minimum Range		
SMCEVA(N,T)	A-Coefficient for Superellipse		
SMCEVB(N,T)	B-Coefficient for Superellipse		
SMCEVN(N,T)	Exponent for Superellipse		
SMCPHT(N,T,3)	Probability of Hit Table		
SMCPGP(N)	Probability of Hit in Gap		
SMCGTP(N)	SAM Guidance Type (1=Command All-The-Way, 2=Home All-The-Way, 3=Mid-Course Guidance)	I*2	
SMCTOF(N,T,3)	Time of Flight Table corresponding to P-Hit		
SMCTGP(N)	Length of Time in Gap		
SMCWH(N)	Warhead Type (0=Conv, Positive is index to NWCHAR		

Table 3-9. FORTRAN Common Variables (continued)

/SNCHAR/

Sensor Characteristics

N = Sensor Type, 20 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
SNCDR(N)	Range Beta (Clear environment detection range on 1 sq meter target)	R*4	F-11
SNCSS(N)	Search Sector, Radians		
SNCIR(N)	Instrumented Range		
SNCDRA(N)	Range Alpha (Burnthrough range on 1 sq meter target with SSJ of 1 Watt per MHz)		
SNCSLR(N)	Sidelobe Ratio (Average arithmetic ratio of main beam to sidelobes)		
SNCJDW(N)	Jamming Bandwith		

Table 3-9. FORTRAN Common Variables (continued)

/TGT/

Arrays for TGTCAP and TGTSAM

N = Number of Red Units, 200 max
M = Combination of Red and Blue Units, 500 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
AIRLST(N)	Red IDs of Known Targets in Forward Air Zone	I*2	
AIRASN(N)	VFID of VF Assigned to AIRLST Entry		
AIRCLK(N)	Time that Confirmation of Target Assignment is Expected (Zero for Unassigned Targets)		0
AIRCNT	Count of Current AIRLST Entries		0
PILCAP(M)	VFID of CAP in Primary Intercept List (PIL)		
PILTGT(M)	AIRLST Index of Target		
PILTIM(M)	Time of Intercept for VF-Target Pair		
PILCNT	Count of Entries in PIL		0
PILDEL	Count of Deletions from PIL		0
SAMLST(N)	Red IDs of All Known Red Missiles, and Red Airplanes that are Inside the Minimum Intercept Range for VF (MININT)		
SAMTGS(N)	Indicator of whether Target in SAMLST is Assigned (1=Assigned, 0=Unassigned)		0
SAMCLK(N)	Estimated Time that Target Will Reach Force Center		
SAMCNT	Number of SAMLST Entries		0

Table 3-9. FORTRAN Common Variables (continued)

/VFSTAT/

VF Status

N = VFID = Blue Unit ID - VFBIAS , 67 max

VARIABLE NAME	CONTENT	DATA TYPE	INITIAL VALUE OR FILE
VFXC(N)	X-Coordinate	R*4	F-21
VFYC(N)	Y-Coordinate		
VFZC(N)	Z-Coordinate		
VFXS(N)	X-Velocity		---
VFYS(N)	Y-Velocity		
VFZS(N)	Z-Velocity		
VFFBR(N)	Fuel Burn Rate		
VFFREM(N)	Fuel Remaining at Time of Last Update		F-22
VFIXC(N)	X-Coordinate of Planned Intercept		---
VFIYC(N)	Y-Coordinate of Planned Intercept		
VFIZC(N)	Z-Coordinate of Planned Intercept		
VFIXST(N)	Tentative X-Velocity		
VFIYST(N)	Tentative Y-Velocity		
VFIZST(N)	Tentative Z-Velocity		
VFSSTT(N)	Tentative Speed Status		
VFTLU(N)	Time of Last Update	I*2	F-21
VFSTAT(N)	VF State		---
VFSST(N)	Speed Status (1=Max Endurance, 2=Max Range, 3=Buster, 4=Gate)		
VFMTYP(N,4)	Missile Types Loaded		F-22
VFMREM(N,4)	Count of Missiles Remaining		
VFTARG(N)	ID of Red Target that VF is Assigned to Engage		0
VFCID(N)	Blue Unit ID of Controller		F-22
VFBIAS	Highest ID Number Assigned to Blue Units Other than VF aircraft		F-21
VFCNT	Number of VFs that Have Entered Game		F-21/22
VFMSTY(N)	Missile Type Selected for Firing		---
VFMSCT(N)	Number of Missiles To Be Fired		
VFCVID(N)	Blue Unit ID of Home Carrier		F-22
VFCPID(N)	Station ID to which VF is Assigned (CPSTAT index)		
VFSEL1(N)	Red ID of Primary Target		0
VFSEL2(N)	Red ID of Secondary Target		0
VFFRST(N)	Firing Status (0=No Target, 1=Reattack, 2=Wait for Hit or Miss But No Reattack)		0
VFREF(N)	Refuelled Flag (T=Has Been Refuelled)	L*1	F
VFNUCF(N)	Nuclear Weapon Flag (T=Nuc is Preferred)		F
VFSELF(N)	Awaiting Response to Self-Assign Request		F

3.6 PROGRAMMING CONVENTIONS

3.6.1 FORTRAN Conventions

1. FORTRAN IV FEATURES, IBM FORTRAN IV (H Extended) was used. The program includes the following FORTRAN IV features not in ANS FORTRAN:

-- Data types INTEGER*2 and LOGICAL*1 are used in common arrays to reduce memory requirements

-- The END parameter is used in READ statements in Subroutine INIT to process end-of-file conditions on variable length data sets.

-- Generalized subscripts are used, including function references.

-- IMPLICIT declarations are used in some subroutines to extend the implied integer variable names to initial letters other than I through M.

-- Some initial data values appear in explicit specification statements.

-- Lengths of variables and arrays are frequently defined as part of type specifications. Data types include REAL*4, INTEGER*4, INTEGER*2, and LOGICAL*1.

-- NAMELISTs were used as an easy means of providing output for program testing.

2. GPSS INTERFACE. All FORTRAN calls from GPSS go through program HLPRTN. This is necessary in order to link all FORTRAN programs together, to permit the programs to share common data. This organization also has the advantage of enabling the FORTRAN - GPSS interface to be restricted to HLPRTN.

3. SUBROUTINE ARGUMENTS. Although data type INTEGER*2 is often used in Common data to reduce memory requirements, integer data are always passed as INTEGER*4 in subroutine calls, to insure type matches between programs.

4. COMMON BLOCKS. Common data have been organized into logical blocks based primarily on the principal subscript for arrays within the block. For example, data indexed on Blue unit ID are separated from data

indexed on Blue aircraft type or Red unit ID. To enable faster recognition of what the data category is and what the subscript should be, identifying two or three letter prefixes are applied to all or most of the arrays in each common block.

Other divisions of common data are based on whether the data represents what is actually happening or represents one player's state of knowledge about what is happening. For example, the command center maintains status data on Blue units based on messages he receives.

5. MAINTENANCE OF COMMON DATA. In order to ease the problem of updating common data specifications that appear in many different programs and to insure that identical specifications appear in each program, common data specification blocks are maintained as separate data sets. Each data set includes a header line containing the label for the common block and the date it was last updated. The data types and data lengths for each variable or array is explicitly specified preceding the COMMON statement. Whenever a common block is updated, the old specification is deleted in each program it appears in and the new specification is inserted.

6. PROGRAM HEADER. A block of comments are included at the beginning of each source program to describe its function. This header block includes program name, revision number, date of last change, program function, input arguments and output arguments.

7. PROGRAM CLARITY. In order to make the source programs easier to understand, liberal comments have been included and dashed or other separator lines are used to separate logical blocks of code. Spaces have been included within statements to make them more readable.

3.6.2 GPSS Conventions

1. BLOCK SYMBOLS. The first two letters of GPSS block symbols identify the module and the third letter identifies the transaction type that will move through the code. The fourth and fifth characters are numeric and are assigned in sequence to aid in locating blocks being

referenced. The labels thus have the form mmtnn where nn ranges from 01 to 99 and

mm = Module	t = Transaction Type
BS - Blue Scenario	D = Driver Control
CC - Command Center	D = Nuclear Damage
CT - Air Controller	F = VF Control
CV - CV (Carrier)	M = Message
DR - Driver	P = Red Parent
DT - Detect & Track	S = Sensor Copy
PD - Point Defense & Damage Assessment	E = Event
	R = Report
RD - Red Threat	
SS - SAM Ship	
VA - VAW Module	
VF - VF Interceptor	

2. USER CHAIN. An important part of the NADS design is the use of User Chain 1 to hold transactions that are scheduled for future events, and the unlinking of selected transactions to reschedule or terminate them if a current event affects the occurrence of the scheduled future event. In order to do this, transactions must be kept on the user chains, rather than the current events chain (CEC) or future events chain (FEC).

Transactions are kept off the FEC by restricting the use of ADVANCE and generate blocks to the Driver module. Transactions enter the ADVANCE block in the Driver module only when they are unlinked for the next scheduled event.

Coding that might block transactions and hold them on the CEC while another transaction is being processed has been avoided. No PREMPT blocks have been used. Only the Driver and CV modules use the following blocks:

-- SEIZE
-- ENTER

-- GATE (Conditional entry mode)

-- TEST (Conditional entry mode)

3. TRANSACTION PARAMETERS. In some instances, transaction parameters are used for multiple purposes for different types of transactions. However, the major control parameters are restricted to a single function. These include Blue unit ID, Red unit ID, transaction type, event time and event address.

4. SAVEVALUES AND MATRICES. Each savevalue and matrix element is used for a single purpose. Normally they are used for data that is shared among transactions. In some instances they are used as temporary storage for individual transactions in an isolated section of code involving a single event. When a savevalue is used in this way, each transaction stores a value in the savevalue and uses and modifies the value as needed. When the transaction links to the user chain, the savevalue is given up for use by the next transaction moving through the given section of code.

5. PASSING DATA BETWEEN TRANSACTIONS. It is difficult to pass data from one GPSS transaction to another when the savevalue holding the data is subject to change by a third transaction that gets control before the data transfer is completed. The need to pass information to another transaction occurs occasionally in NADs when transactions are split off a parent transaction (or unlinked from a user chain to be modified or rescheduled. Since the driver unlinks transactions one at a time from the user chain for processing, the current events chain (CEC) normally contains only the transaction being processed, transactions unlinked for modification by the transaction being processed, the control transaction in the Driver module (always last on the CEC), and possibly some VF control transactions in the CV module (which do not interact with other transactions). Thus if any transactions are split off they will be processed right away (after the current transaction completes processing by either linking to the user chain or by terminating). In this normal situation the data can be passed to the new transaction without problem.

However if the transaction being processed is one of a group of sensor copy transactions that have been unlinked for recomputation of event times, the CEC will be holding all of the unlinked transactions. Then if a new transaction is split off it will probably not be the next one processed and thus will be unable to obtain data values being passed via savevalues.

This type of situation has been taken care of in one of several ways:

- o By temporarily raising the priority of the parent transaction before splitting off the new transaction and then entering a BUFFER block so the new transaction will be able to immediately obtain the desired value from the savevalue.
- o By saving the contents of suitable parameters in the parent transaction and then storing the data into these parameters before splitting off the copy, so that the data will be available to the copy transaction, no matter when it gets processed. The original contents of the parent's parameters are then restored.
- o By routing the parent transaction into the code that needs the savevalue and letting the copy take over as the parent. This technique is used only when the parent does not need any data from savevalues and when it doesn't matter which code is executed first.

None of the above techniques can be used to enable a transaction to pass data to a transaction that it unlinks from the user chain. SAVEVALUEs can be used in this case by using the PRIORITY block with the BUFFER option to set the priority of the unlinking transaction below that of the transaction to be unlinked and to clear the CEC of all tranactions of the same or higher priority as the unlinking transaction. When the UNLINK block is used to unlink the transaction it has the highest priority on the CEC and will be processed after the transaction being currently processed has moved as far as possible. Data can now be processed using

SAVEVALUEs with confidence. Note that the PRIORITY block without the BUFFER option does not change the processing of the current transaction. If savevalues cannot be used, the only means of "passing information" is by the program location to which the unlinked transaction is sent. For example, if a sensor copy is to be terminated it is sent one place and if detection times are to be recomputed it is sent somewhere else.

6. DATA WORD LENGTHS. Where possible, byte and halfword parameters, savevalues and matrices are used instead of fullword as a means of reducing core requirements. An exception to this is that savevalues and matrices that are referenced by FORTRAN are restricted to halfword or fullword.

7. SYMBOLIC NAMES. In the GPSS program, symbolic names are used to reference most GPSS entities (e.g., savevalues) instead of numbers. However, the symbolic names are all assigned specific numbers via EQU statements. This eliminates possible mismatches when a FORTRAN program references a savevalue or other entity by number. The use of EQU statements also eliminates possible confusion about desired overlap with other numbered entities.

8. RELATIVE ADDRESSING. The use of relative addressing (e.g., TRANSFER, SSS20+1) has been kept to a minimum. In most instances blocks that are referenced are given symbolic names. An exception occurs when a transaction is to transfer to the next block (which is referenced as "*+1") and the transfer will not be affected by coding changes. Another exception occurs in macros. Labels cannot be used within a macro, unless they are arguments of the macro. Since this is inconvenient, some of the macros use relative addressing that must be very carefully modified if any coding changes are required.

9. COMMENTS. For ease of understanding, the GPSS code is freely commented. Comments for individual lines of code begin in column 45. In addition, sections of code are commented where it appeared to be useful.

3.6.3 GPSS to FORTRAN Address Conversion

GPSS data is stored in arrays that, in most cases, are incompatible with the FORTRAN array structure. It is necessary to convert from GPSS entities to FORTRAN subscripted arrays in order to access GPSS data within a FORTRAN subroutine. The algorithms for making the conversions are described on pages 165-168 of the GPSS Users Manual.* This section describes how the algorithms have been applied in NADS.

The specific array names to be used in FORTRAN to reference GPSS entities are shown in Table 3-10 (Table 12-1 from GPSS Users Manual). A single subscript must be computed for each entity/array element. For example FORTRAN variables

ISAVEF(J) references a fullword savevalue

IMAXBH(J) references a halfword matrix element.

The computation of the subscript J is described below.

SAVEVALUES

For savevalues, facilities, storages, queues, logic switches, tables and user chains, the subscript is defined as

$$J = K * (N-1) + L$$

where K and L are constants for each entity type and N is the index number of a specific entity type, for example queue 8 or halfword savevalue 2.

Fortunately, in the case of savevalues, K and L both equal 1, making J = N. Thus it is unnecessary to use this formula in referencing savevalues. Even for the other entities listed above, the computation can be simplified by substituting values for K and L in the formula. This reduces the conversion of some sample entities to the following:

---

*General Purpose Simulation System V User's Manual, IBM Publication SH20-0851-2.

Table 3-10. GPSS Reference Information for FORTRAN HELPC Routines (Table 12-1 From GPSS Users Manual)

Entity	Attribute	Indexing K	Constants L	Reference Word	FORTRAN Mode	FORTRAN Length (bytes)
SAVEVALUE (Fullword)	Contents	1	1	ISAVEF	INTEGER	4
SAVEVALUE (Halfword)	Contents	1	1	ISAVEH	INTEGER	2
SAVEVALUE (Floating Point)	Contents	1	1	FSAVEL	REAL	4
Facility	Cumulative time integral (F3)	7	2	IFAC	INTEGER	4
	Clock time of last status change (F4)	7	3	IFAC	INTEGER	4
	Entry count (F9)	7	6	IFAC	INTEGER	4
Storage	Current contents of Storage (S1)	11	1	ISTO	INTEGER	4
	# of available units in Storage (S2)	11	2	ISTO	INTEGER	4
	Cumulative time integral (S3-S4)	11	3	FSTO	REAL	4
	Last clock time Storage changed status (S5)	11	5	ISTO	INTEGER	4
	Entry count (S6)	11	6	ISTO	INTEGER	4
	Max. Storage, contents (S7)	11	7	ISTO	INTEGER	4
Queue	Last clock time Queue changed status (Q1)	8	1	IQUE	INTEGER	4
	Total entry count (Q2)	8	2	IQUE	INTEGER	4
	# of zero delay entries (Q3)	8	5	IQUE	INTEGER	4
	Cumulative time integral (Q4-Q5)	4	2	FQUE	REAL	8
	Current contents of Queue (Q6)	8	6	IQUE	INTEGER	4
	Maximum contents (Q7)	8	7	IQUE	INTEGER	4
Logic Switch	Logic Switch status (L1)	3	1	ILOG	INTEGER	2
Table	Sum of arguments in Table (D1)	8	1	FTAB	REAL	8
	Sum of squared arguments (D2)	8	2	FTAB	REAL	8
	Sum of weighted values (D3)	8	3	FTAB	REAL	8
	Sum of weighted squared values (D4)	8	4	FTAB	REAL	8
	# of entries (D6)	16	10	ITAB	INTEGER	4
User Chain	# of transactions on User Chain (U3)	12	3	IUSE	INTEGER	2
	Max. # of transactions on User Chain (U4)	12	4	IUSE	INTEGER	2
	Total # of transactions on User Chain (U5)	6	3	IUSEF	INTEGER	4
	Clock time of last status change (U6)	6	4	IUSEF	INTEGER	4
	Cumulative time integral (U7-U8)	3	3	FUSE	REAL	8
MATRIX (Fullword)	MSAVEVALUES, i.e., given row and column	6	1	(IMAX to calculate subscript J)	INTEGER	4
				(IMAXB to reference given msavevalue)	INTEGER	4
MATRIX (Halfword)	MSAVEVALUES, i.e., given row and column	6	1	(IMAXH to calculate subscript J)	INTEGER	4
				(IMAXBH to reference given msavevalue)	INTEGER	2
MATRIX (Floating Point)	MSAVEVALUES, i.e., given row and column	6	1	(IMAXL to calculate subscript J)	INTEGER	4
				(FMAXBL to reference given msavevalue)	REAL	4

GPSS Entity Type	FORTRAN Reference For Entity Number N
Savevalue (Fullword)	ISAVEF(N)
Savevalue (Halfword)	ISAVEH(N)
Queue - Current Contents	IQUE(8N - 7)

MATRICES

The conversion algorithm for GPSS matrices is more complex. A function subprogram called MATRIX has been developed to compute the subscript J for GPSS halfword matrices, which is the only type currently in use in NADS. The function is referenced as MATRIX (N,K,1,IMAXH) where

N = GPSS halfword matrix number

1 = RED	4 = MSG	7 = RUISN
2 = BLUE	5 = BUDMG	8 = ALTIM
3 = ACCAR	6 = RUDMG	9 = CVCAR

K = matrix row (i.e., ID of Blue or Red unit, message number or index of damaged unit)

1 = Matrix column (see matrix definitions in GPSS Program)

IMAXH = GPSS system name that must be passed to MATRIX

A FORTRAN program can then store the tracking capacity of blue unit 5 for use by GPSS by referencing

```
IMAXBH(MATRIX(2,5,2,IMAXH))
```

since tracking capacity is stored in Column 2 of GPSS Matrix BLUE (halfword matrix 2).

3.7 PHYSICAL CONVENTIONS

3.7.1 Geometry Conventions

The NADS geometry is based on a right-handed cartesian coordinate system. The X axis is positive East, Y is positive North, and Z is positive upward. The origin is at the sea surface. The surface is assumed to be a horizontal plane. Radar lines of sight, consequently are curved. The curvature must be taken into account in computing horizon distances, but in many other contexts it can be ignored. It is not considered in the computation of slant ranges, for example.

There is no specific constraint on the location of the origin relative to the positions of the Blue Units that comprise the CV Battle Group. In the tactical logic of NADS the origin is presumed to be the "Force Center," which implies that it should be in the general vicinity of the defended point, the centroid of the positions of the carriers and AAW escort vessels. If it were somewhere outside the ship formation the program would still run, but some of the tactical logic would be irrational.

3.7.2 Physical Units

The general practice in the NADS is to use traditional English units for input data, for the convenience of most users. To minimize the error risks and computing overhead of frequent conversions between subroutines, nearly all the internal computations are in the metric (SI) system.

An exception is in the area of nuclear weapons effects where some previously developed subroutines were adapted with minimal changes. The unit conversions within subroutine NUCLER and its subordinate routines are discussed in more detail in Section 3.7.3. With that exception, the following material applies to all of NADS.

In GPSS the only physical variable is the simulation clock time, C1. It is a positive integer, ranging from one second as the earliest possible event time, to a maximum of 32,767 seconds (about 9.1 hours).

In FORTRAN, subroutine INIT reads the input data files and makes the necessary conversions to internal units before the values are stored as the corresponding common variables. FORTRAN computations are frequently used to obtain future event times for GPSS. When the real mode is used, the resulting event times are truncated for use in GPSS. Table 3-11 shows the standard NADS units.

Table 3-11. NADS Physical Units

QUANTITY	INPUT UNITS	INTERNAL UNITS
Time	Minutes for most data; seconds for data that are typically less than one minute	Seconds
Speed	Knots	Meters per second
Position (X,Y)	Nautical miles	Meters
Altitude (or Height)	Kilofeet (including ships' antenna heights)	Meters
Burn Rates	Pounds per hour	Kilograms per second
Angles	Degrees	Radians

3.7.3 Unit Conversions Within NUCLER and Its Subprograms

NADS carries position data in meters, velocities in meters per second, time in seconds, yields in kilotons.

NUCLER reads positions and velocities in NADS units then coverts positions to kilometers and velocities to km/sec. Times and yields remain as sec and KT.

BLINT is called by NUCLER, and it retains the km, sec, KT dimensions.

RSHK is used by BLINT. It requires input in km and KT and returns result in km.

MATM62	is used by RSHK (and other places). I.. input is altitude in kilometers and a conversion flag that selects either metric or English units for output. If the flag = 0, then MATM62 returns: -- pressure in dynes per square centimeter -- sound speed in centimeters per second -- density in grams per cubic centimeter -- temperature in degrees Kelvin If the flag = 1, MATM62 returns: -- pressure in pounds per square inch -- sound speed in feet per second -- density in slugs per cubic foot -- temperature in degrees Kelvin
TSHK	is used by NUCLER. Inputs in km and KT require no conversions. Returns seconds.
BLAST	is called by NUCLER with km, sec, KT arguments. BLAST converts to kilofeet and megatons for internal use and for passing to most of the subprograms that it calls upon. BLAST returns psi, psi-seconds, and feet per second, for direct comparison with vulnerability criteria, which are input and stored in the same units.
SCALE	is called by BLAST with kft and MT arguments. SCALE defines scaling factors for fitting to 1KT, sea-level data. The outputs are stored in Common/SCALT/.
RTPP	is called by BLAST with scaled altitude. Returns a scaled horiziontal range.
PFA	is called by BLAST or by PMS with scaled range. Returns pressure in psi for 1 KT.
PMS	is called by BLAST with scaled range. Returns pressure in psi for 1 KT.
PULSE	is called by BLAST, using range and yield in Km and KT that it received from NUCLER. It returns impulse in psi-seconds.
VPARTS	is fundamentally dimensionless. It resolves an input velocity in any units (ft/sec in this case) into two components of the same unit. It uses XYZ positions whose dimensions cancel and XYZ velocities whose dimensions cancel.
RHOX	is called by NUCLER with km arguments. It returns an air mass integral in grams per sq centimeter.

FENV is called by NUCLER with air mass integral in gm/cm^2 and range in km. It returns:

- -- Gamma Peak Dose Rate in Rads(Si) per second
- -- Neutron Fluence in Neutrons per cm^2
- -- Thermal Fluence in Calories per cm^2

3.8 PROGRAMMING ESTIMATES

Table 3-12. GPSS Space Requirements

Entity	Estimated Count	Space Estimate (Bytes)* Basic	Common
Blocks	2,100	25,200	33,600
Transactions (Active)	10,000	160,000	272,000
Facilities	5	140	0
Storages	0	0	0
Queues	175	5,600	0
Logic Switches	2	12	0
Tables	0	0	0
Functions	16	512	1236
Variables	18	864	772
Fullword Savevalues	19	76	0
Halfword Savevalues	35	70	0
Byte Savevalues	5	5	0
Floating Pt. Savevalues	0	0	0
User Chains	2	48	0
Groups	1	4	36
Boolean Variables	27	864	2,156
Halfword Matrices	9	216	4,834
Byte Matrices	1	24	150
TOTALS		193,635	314,784

*Based on Table 20-1 (Pg. 324) GPSS Users Manual

Table 3-13. Summary of Estimated Space Requirements

Usage	Estimated Space (Bytes)
GPSS Entities (See above)	510K
GPSS System Software	38K
FORTRAN Common and Program	370K
TOTAL	918K

Table 3-14. Maximum Counts Associated with Arrays

BLUE

VF Aircraft	- 67	
VAW Aircraft	- 6	Combined
Ships	- 60	Max=60
Aircraft platform types	- 10	
VF missile types	- 6	
LAR's (missile types x speed)	- 18	
Sensor types	- 20	
Ship platform types	- 10	
SAM types	- 10	
Illuminators/Ship	- 8	
Fire control channels/ship	- 24	
Fixed Cap stations	- 15	

RED

Aircraft	- 60	Combined
Missiles (SSM, ASM)	-150	Max=200
Aircraft platform types	- 20	
Missile types	- 10	
Jammers aircraft	- 6	
Aircraft flight nodes	- 20	

RED & BLUE

Nuclear warhead types	- 10
Total nuclear bursts	-150

Table 3-15. Estimated Maximum Number of Active Transactions

Transaction Type	Count
Red Parents	175
Sensor copies (175 * (20VF + 35 ships)	9,625
Messages	83
VF Control	67
Nuclear Damage	40
Miscellaneous (Control, timer, external surveillance)	10
TOTAL	10,000

4. DRIVER MODULES

4.1 THE DRIVER MODULE

The Driver Module initiates the processing of a simulation run and performs the following functions:

- o Initializes data values
- o Causes the next event in the game to occur
- o Ends the game

The Driver generates and processes two transactions - a Control transaction and a Timer transaction.

4.1.1 Control Transaction

The processing of the Control transaction is shown in Figure 4-1. This transaction will be the first one generated in the game in order to perform the initialization of all GPSS and FORTRAN data arrays. Further processing of this transaction is then delayed until the initial Red Threat and Blue Scenario transactions are generated and linked to the user chain.

When processing of the Control transaction continues, it moves into the event step loop, where it stays for the remainder of the game. It unlinks the first transaction from the user chain, which is the next event scheduled to occur, since the user chain is sequenced by event time and priority. Further processing of the Control transaction is delayed until the transaction just unlinked can be processed or, in effect, until the scheduled event can occur. At this time the Control transaction loops back and again unlinks the first transaction from the user chain.

The Control transaction remains on the current events chain throughout the game. It is never linked to the user chain or the future events chain.

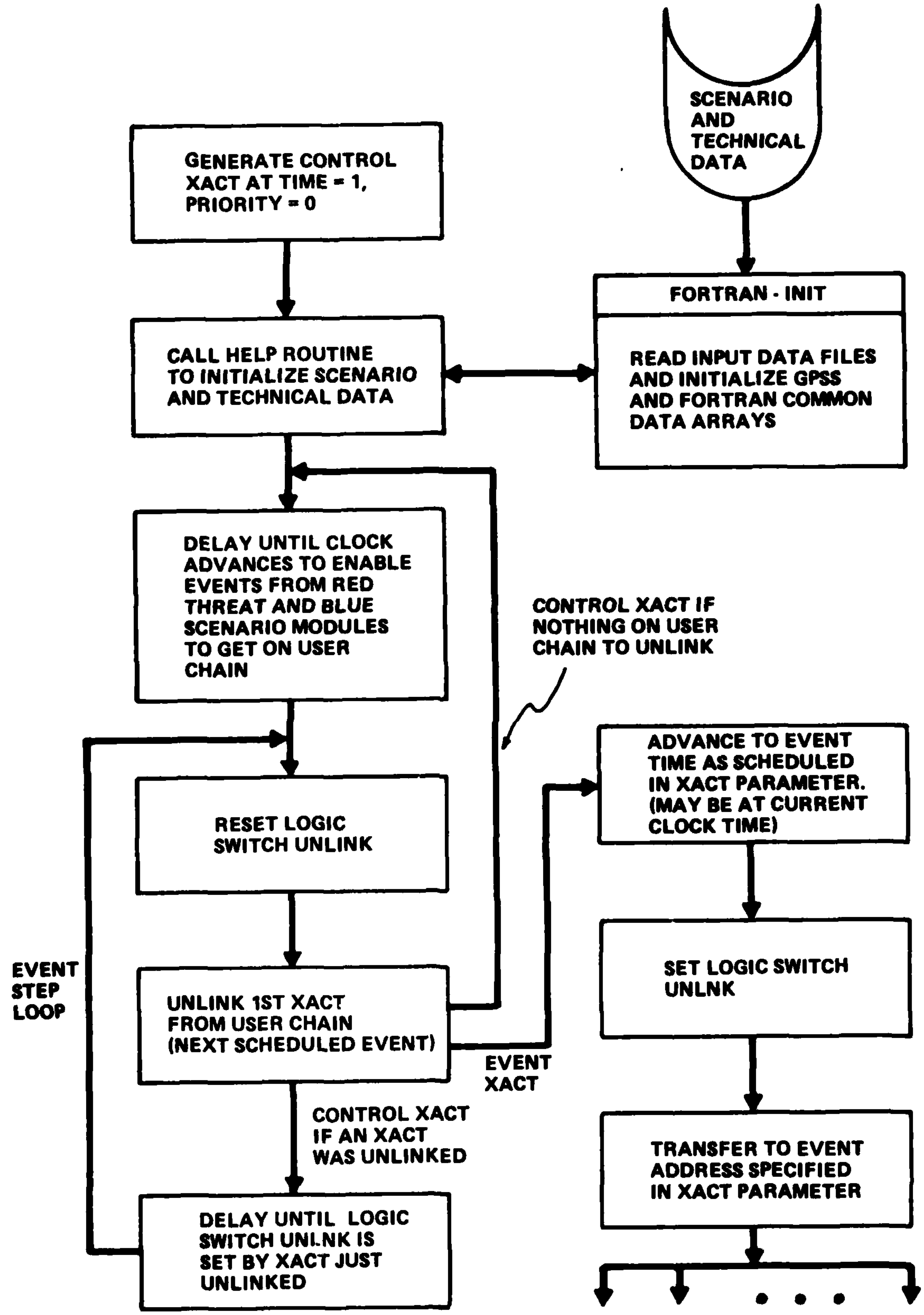


Figure 4-1. Driver Module (Control Transaction)

4.1.2 Timer Transaction

The processing of the Timer transaction is shown on the flowchart in Figure 4-2. This transaction will not be processed until the game is scheduled to end. At this time the transaction will cause all special formatted reports to be produced prior to terminating the game. The next game, if any, will then be initiated.

The Timer transaction will be held on the future events chain by GPSS until the clock advances to the scheduled end of game. At that time it will be put on the current events chain for processing.

4.2 RED SCENARIO MODULE

The Red Scenario (or Threat) Module generates and processes transactions representing the following elements of the Red attack:

- o Red Aircraft (Bombers, Jammers, Recce, and Escort Fighters)
- o Red Missiles (Sea Launched and Air Launched)

The aircraft and sea-launched missile transactions are generated from the input scenario. The air-launched missile transactions are split from the Red bomber transaction at the time the weapons are launched. This launch time is tentatively defined in the scenario, but may be delayed by the events of the game.

The general processing scheme is described below. The flowcharts for the two types of Red threat transactions are shown in Figure 4-3. The FORTRAN supporting routine, RNODE, is shown in Figure 4-4.

4.2.1 Red Aircraft

One Red Parent Aircraft transaction is generated for each enemy bomber and jammer aircraft specified in the input scenario. The events associated with the aircraft are the nodes of the flight path, which may include the following:

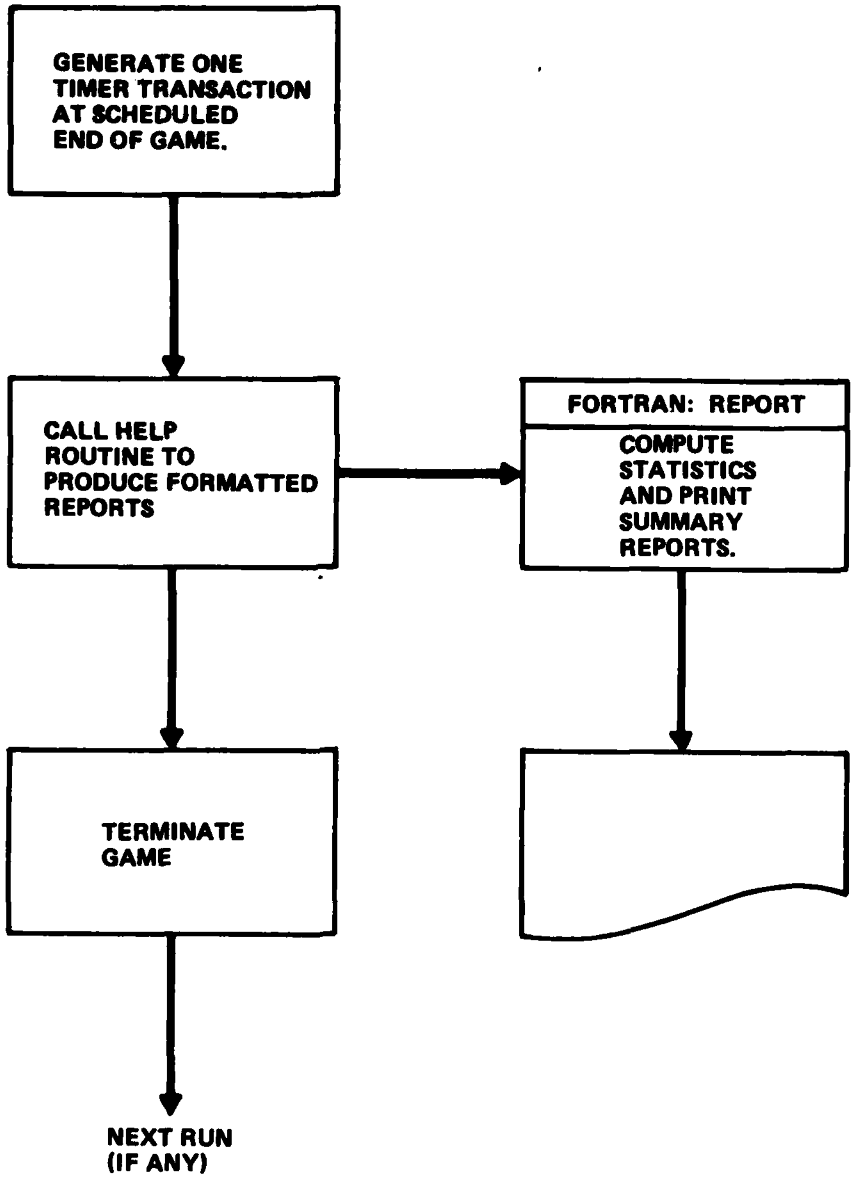


Figure 4-2. Driver Module (Timer Transaction)

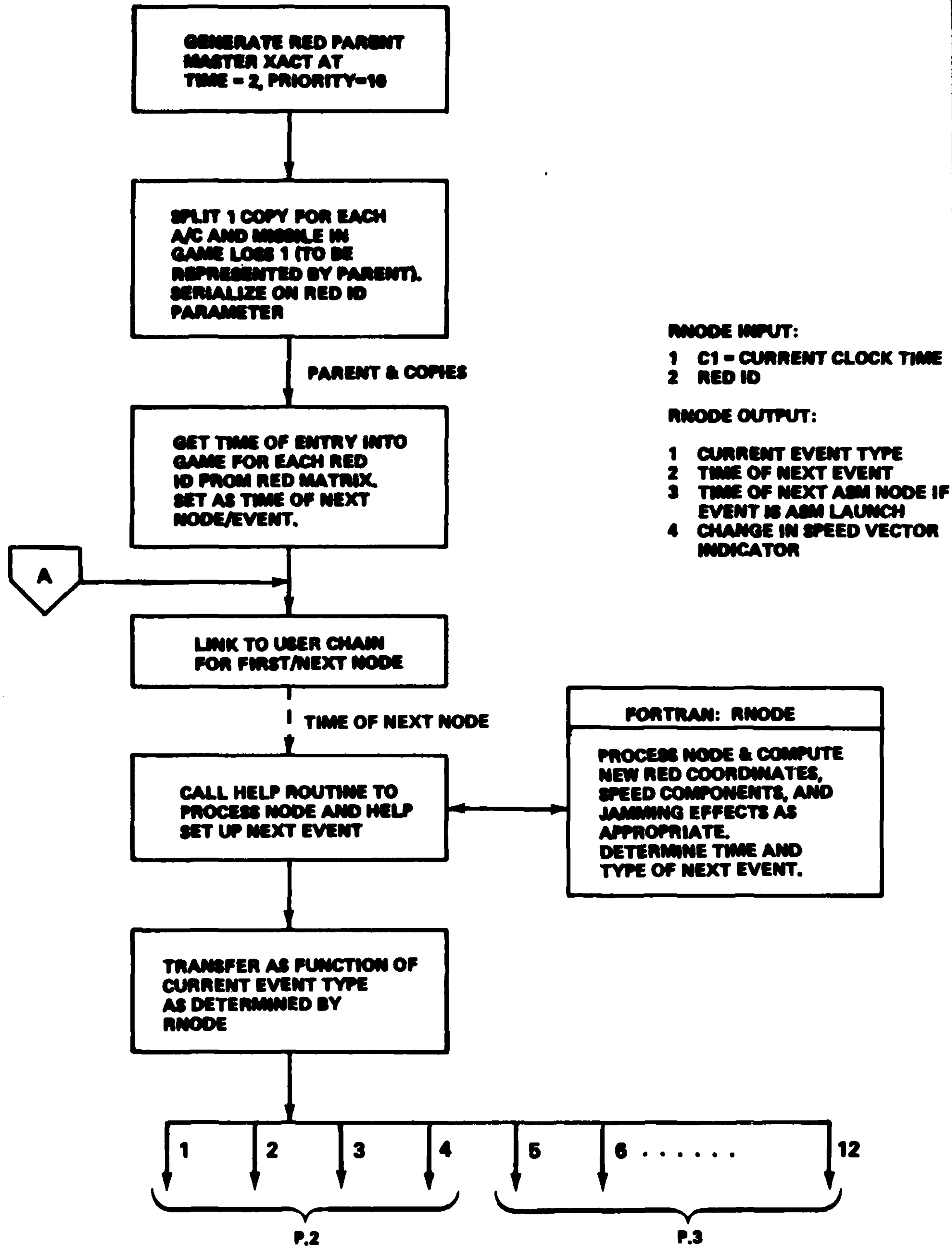


Figure 4-3. Red Threat Module (GPSS) (Page 1 of 3)

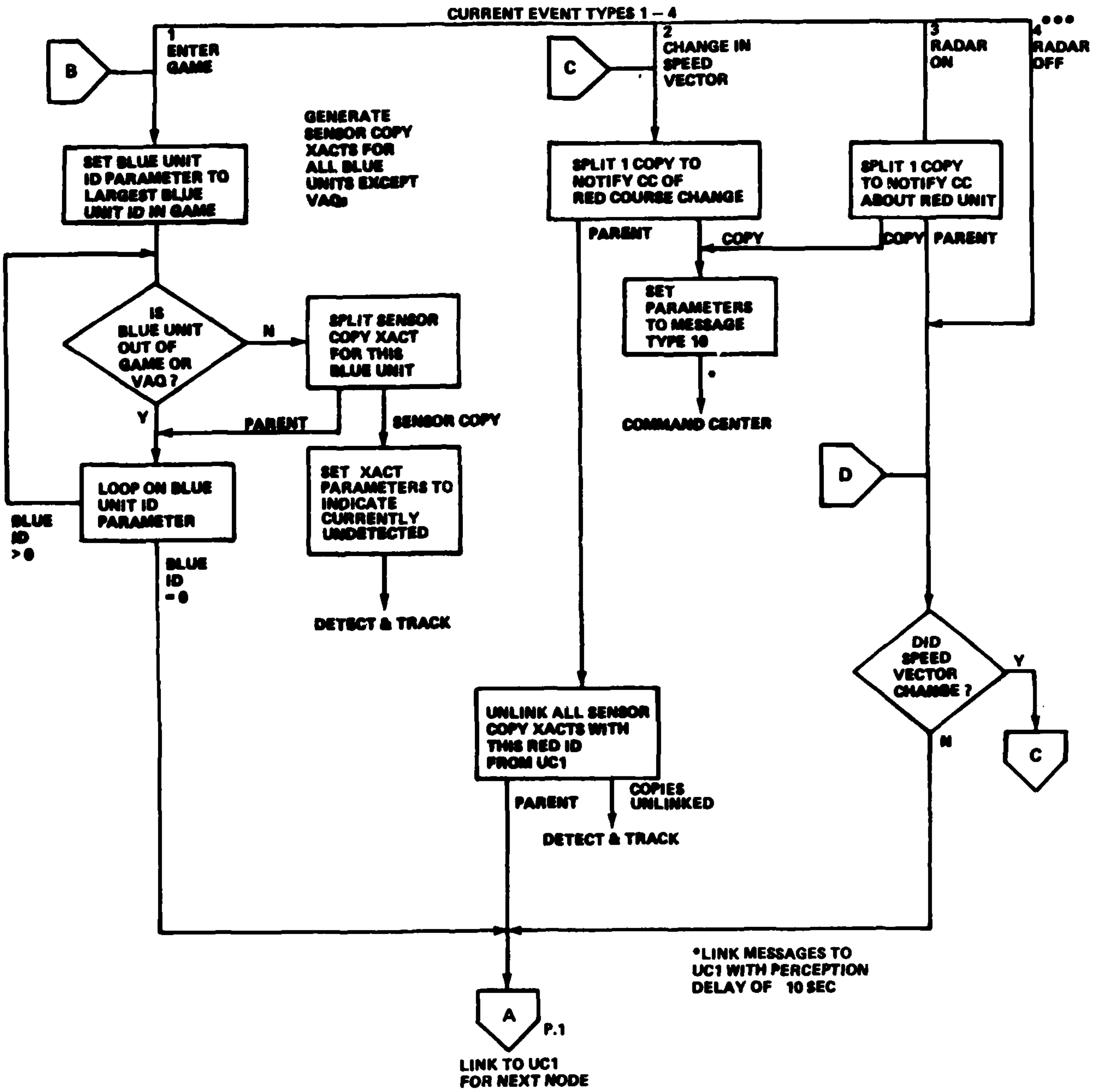


Figure 4-3. Red Threat Module (GPSS) (Page 2 of 3)

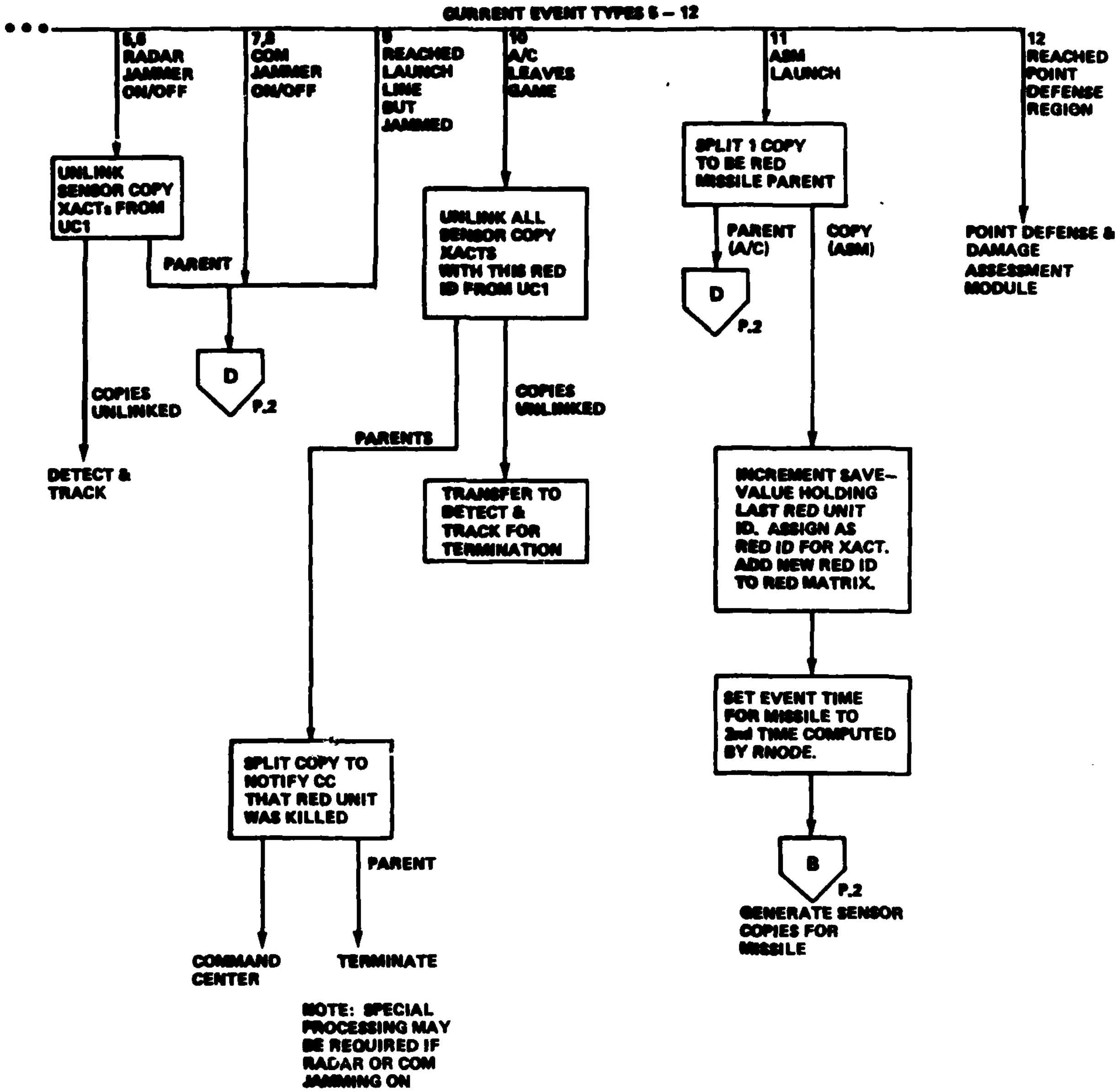


Figure 4-3. Red Threat Module (GPSS) (Page 3 of 3)

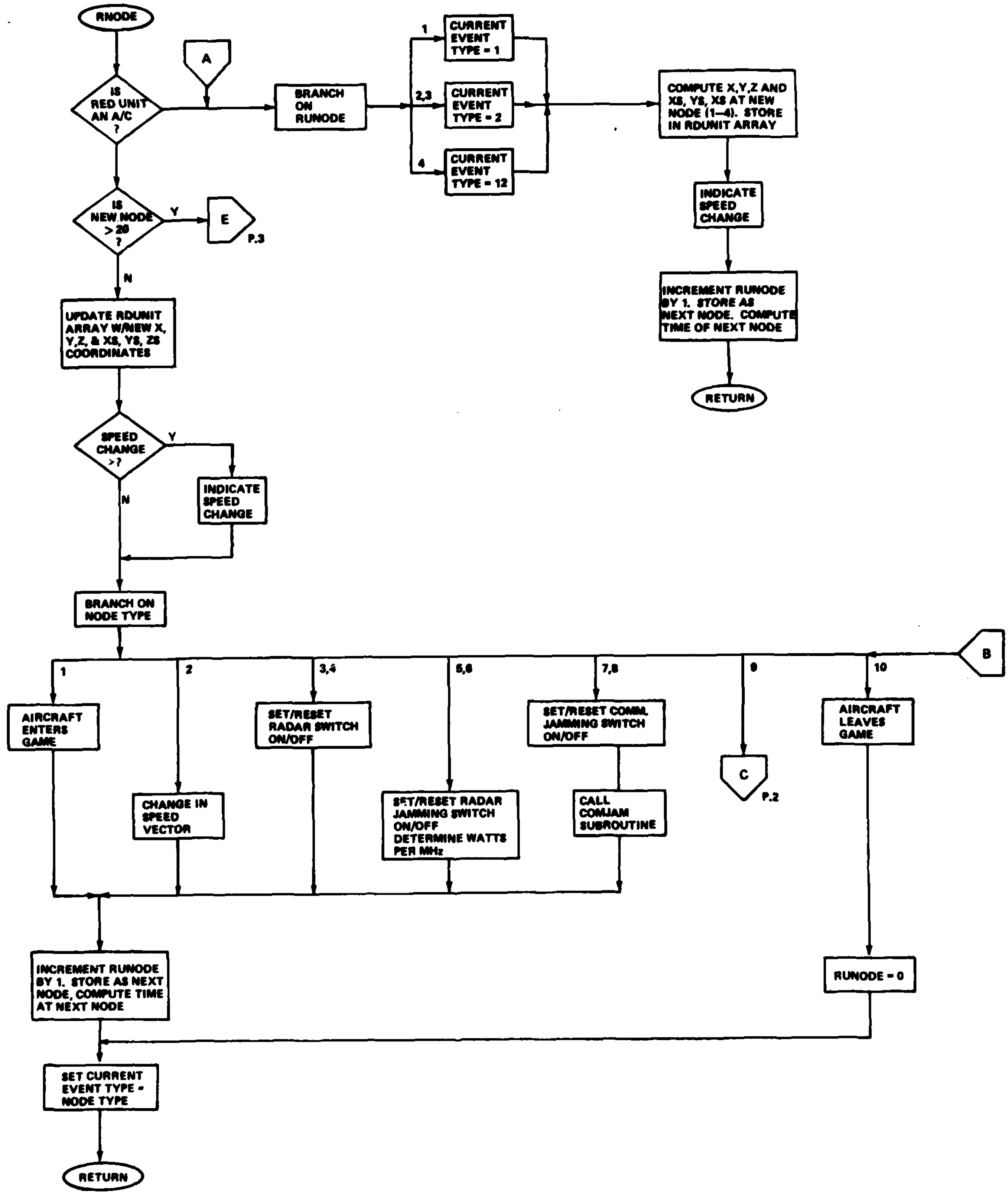


Figure 4-4. Subroutine RNODE Red Threat Module
(Page 1 of 3)

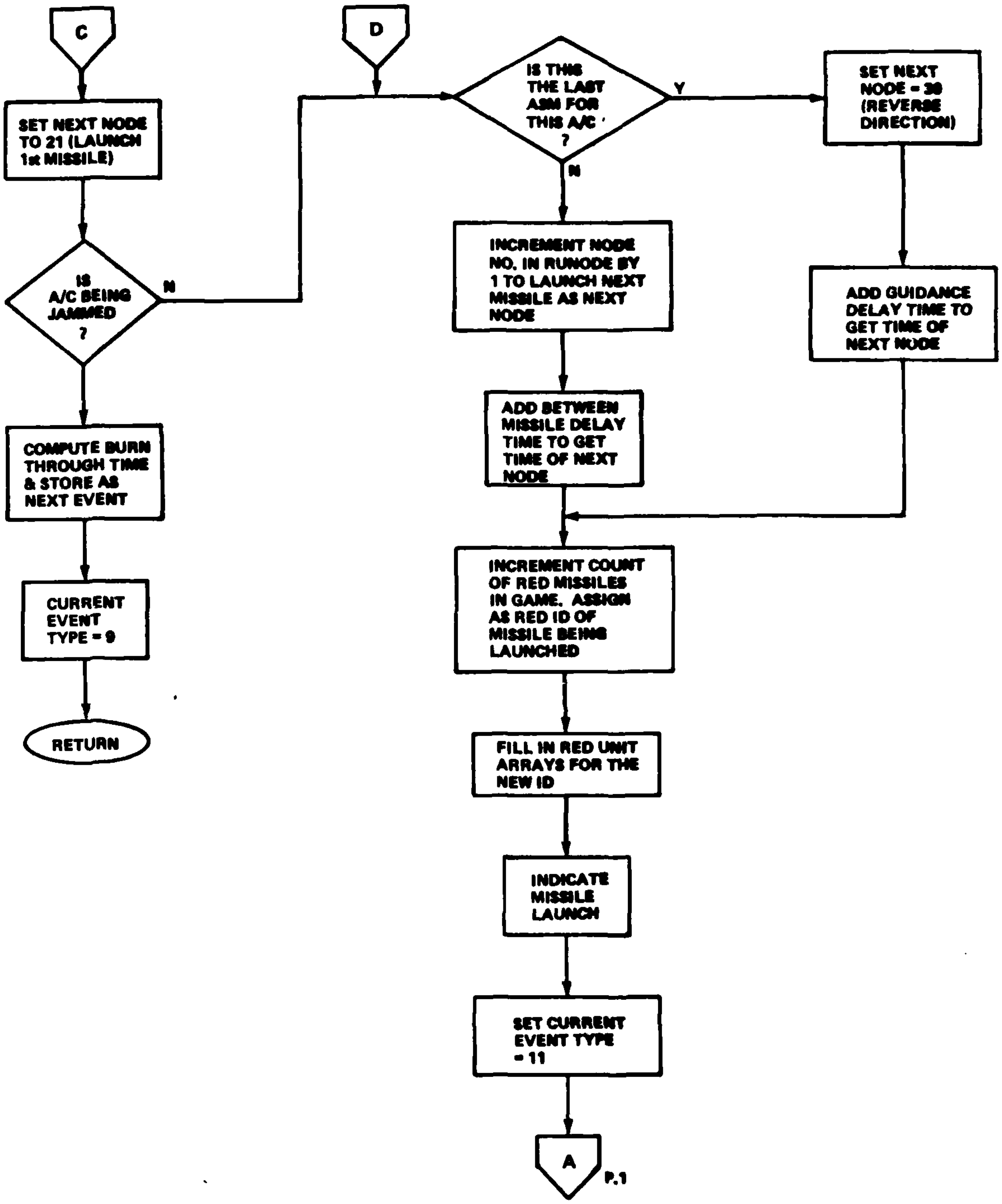


Figure 4-4. Subroutine RNODE (Cont.) Red Threat Module (Page 2 of 3)

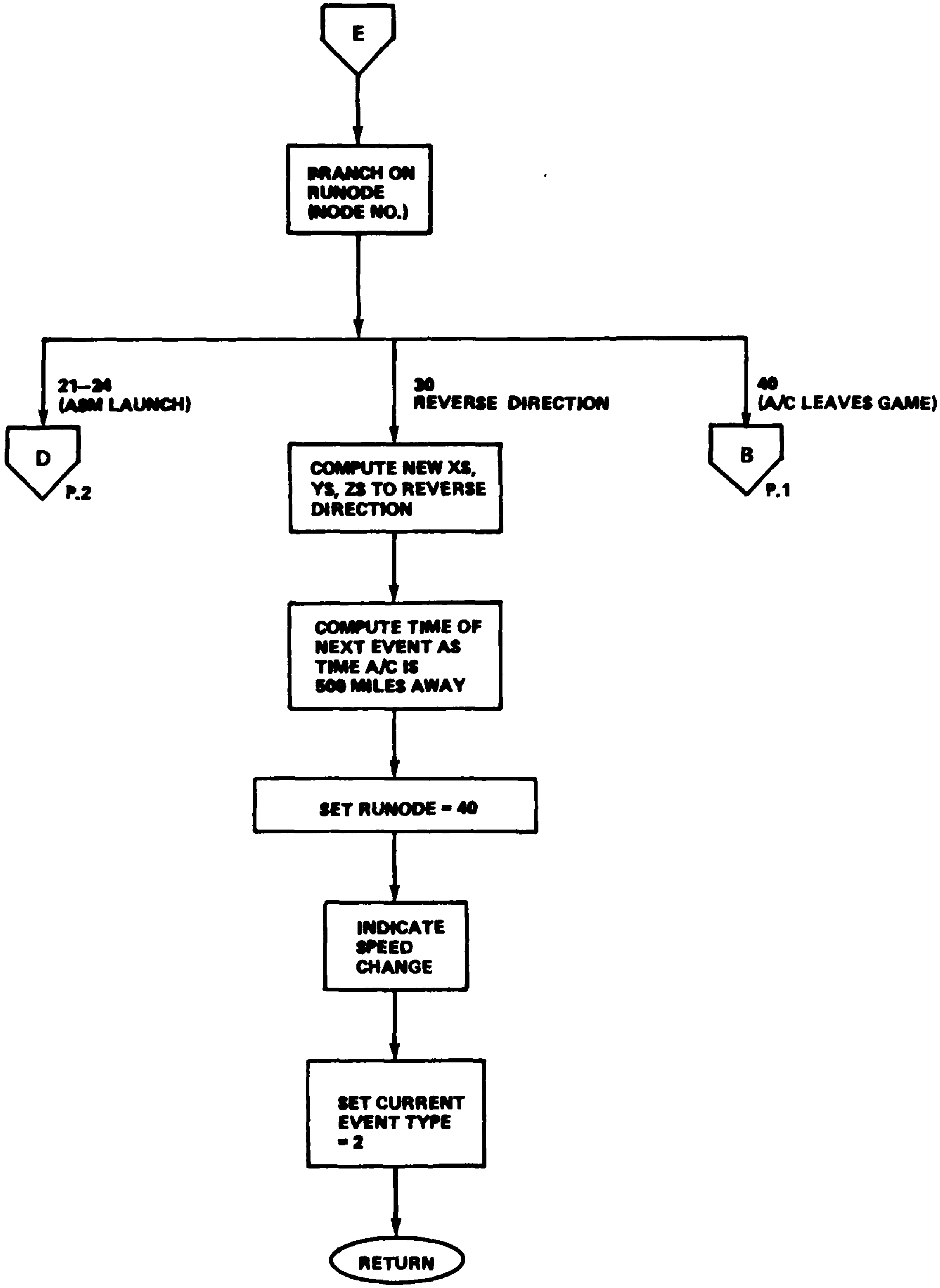


Figure 4-4. Subroutine (RNODE (cont.) Red Threat Module (Page 3 of 3)

- o Change in velocity vector (Heading, speed, rate of climb)
- o Radar on/off Bomber only)
- o Jammer on/off (SOJ or SSJ)
- o Missile launch (Bomber only)
- o Removal of aircraft from game

Red aircraft Parent transactions are put on the user chain initially for the first node of the flight path. Copies of the transactions are split off for each Blue sensor and sent to the Detect and Track Module to compute detection times and set up detection events. Following the processing of a node, the Parent transaction is modified, via the event time and event address parameters, to simulate the next node of the flight path. It is then relinked to the user chain. Note that the event address will point to the code corresponding to change in velocity vector, radar on/off, etc., depending on the next node specified in the input scenario.

4.2.2 Red Missiles

One Red Parent Missile transaction is generated for each enemy sea-launched missile in the input scenario. The processing is similar to that for Red aircraft, except that the intermediate nodes of the flight path are computed from the missile characteristics data rather than scenario defined. If the missile is not shot down, the transaction will move to the Terminal Defense and Damage Assessment Module for processing at the time computed for the warhead burst.

Air-launched missile transactions move from the aircraft segment to the missile segment for processing after they are split from the aircraft parent. They are processed similarly to the sea-launched missiles, although the specific computations are different.

4.3 BLUE SCENARIO MODULE

The Blue Scenario module initializes and processes a number of different types of transactions, which are used to simulate certain events associated with the Blue forces. These events are only those that are prescheduled and are independent of the Red attack within the game. The following types of events are included:

- o Aircraft Initialization
- o Command Center Decision
- o External Surveillance Message Receipt

These event transactions are maintained on the GPSS user chain 1 along with other scheduled events. They are unlinked by the Control transaction when the event is scheduled to occur.

Figures 4-5 and 4-7 describe the processing of the Blue Scenario transactions. This processing follows the general pattern outlined below:

1. Generation of one or more transactions to represent a scenario element.
2. Assignment of an event time and event address based on the type of event that is to occur and the time scheduled in the input scenario.
3. Linking of the transaction to the user chain in event time and sequence.
4. In most instances the parent or copy transactions are sent to other modules for processing.

It should be noted that the Blue Scenario Module is composed of a number of different GPSS segments, each of which have their own transaction(s) which move through the segment until they are terminated or are sent to other modules for processing. The advantage of this segmented design is that individual scenaric elements may be added, eliminated or modified, without affecting the other segments of the module.

The general processing of the various Blue scenario transactions was outlined above. The specific characteristics of the processing of individual types is described in the following sections.

Subsystem Failures*

The Model provides a means to simulate the failure of various components of the VAW and VF aircraft and ships. The type and time each failure will occur must be specified in the input scenario.

Figure 4-5 presents a flowchart that describes the processing of subsystem failures. When an aircraft suffers a failure (such as sensor or weapon system failure), the aircraft will return to the carrier. In the case of a ship subsystem failure, the equipment can be repaired. This is simulated byinactivating the sensor detections or other equipment functions for a period of time and then causing the functions to be restored when the scheduled repair time has elapsed.

Aircraft Initialization

One transaction is generated for each Blue fighter in the scenario. At the beginning of a game the transaction event times will be set to the time of arrival on sstation or return to the CV if the aircrft is airborne. Otherwise, no future event is scheduled.

* Not implemented in NADS 4.0

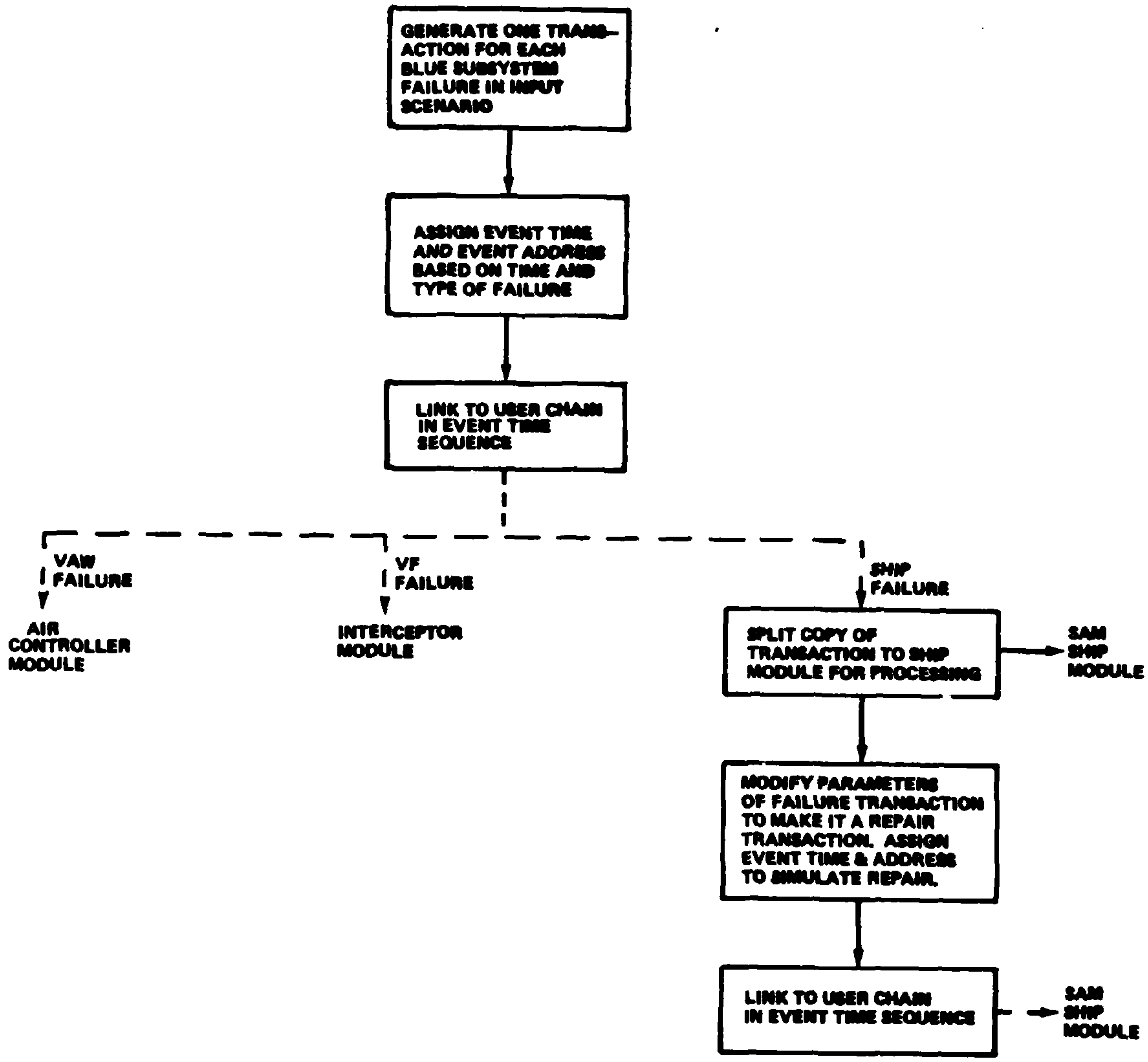


Figure 4-5. Blue Scenario Module (Subsystem Failure)

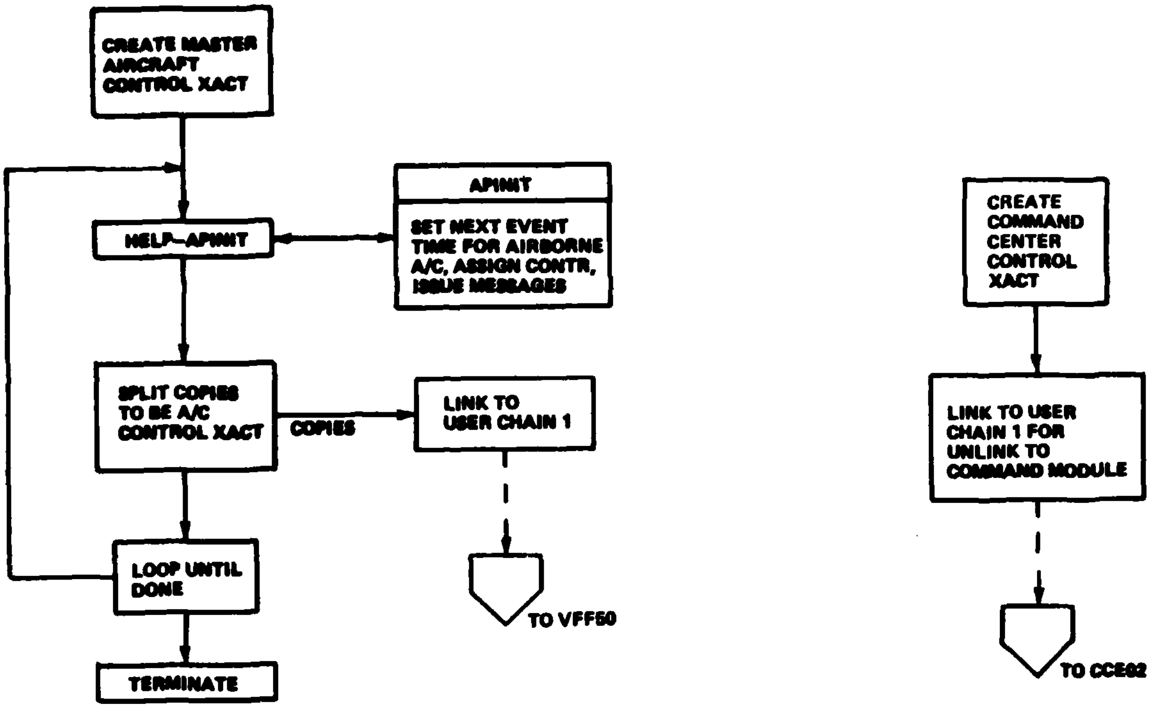


Figure 4-6. Blue Scenario Module (Subsystem Failure)

External Surveillance Messages

To simulate communications linkages with other Blue forces, transactions are generated to represent receipt of external surveillance messages about the Red forces. These messages are specified in the input scenario and are read at the time execution of the last transaction.

External surveillance transactions are put on the user chain and addressed (through the Event address) for processing by the Command Center Module at the time the message is to be received as well as a duplicate which stimulates the reading of the next message in the blue scenario module.

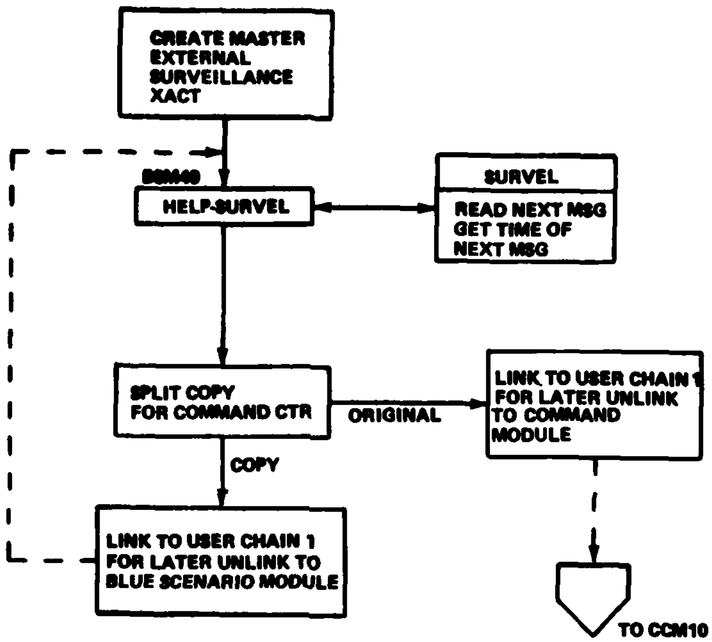


Figure 4-7. Blue Scenario Module (External Surveillance Messages)

5. DETECTION AND TRACKING MODULE

5.1 GENERAL CASE

The detection and tracking functions are performed by VAW aircraft, VF aircraft, and ships. Although passive sensors are part of the defense system, they are modelled implicitly by representing only the effect of the detections.* Each Red/Blue sensor transaction passes through the Detection and Tracking Module. As seen in Figure 5-1, the first block is time of potential detection. This is the time that the target will cross into the detection range if it continues on its present course. No sensors move except those on the VF while flying to CAP station or when on intercept. The potential detection range is a circle centered on the sensor for all fixed sensors, including the VF when on CAP station. The VF sensor detection area is represented as a sector when flying to CAP station or when on intercept. It is assumed that the VF on CAP station will fly a fixed pattern that will result in a detection pattern that can be approximated by a circle. Similarly, the VAW in flying a racetrack pattern will have its detection pattern approximated by a circle.

The size of the potential detection circle (or sector radius) is computed by the FORTRAN help routine illustrated in Figure 5-2. The nominal detection range (Range Beta) of each radar type is among the input data. It represents the radar's clear environment capability against a one square meter target when there is no limitation on operator skill, exhaustion, or division of attention by other targets. The nominal range is corrected for the target's radar cross section, which is also input to get the potential detection range. To qualify as potentially detectable, the target must be within that computed range and also be above the radar horizon. The complications of Figure 5-2 associated with jamming are discussed in Section 5.2.

* Not implemented in NADS 4.0

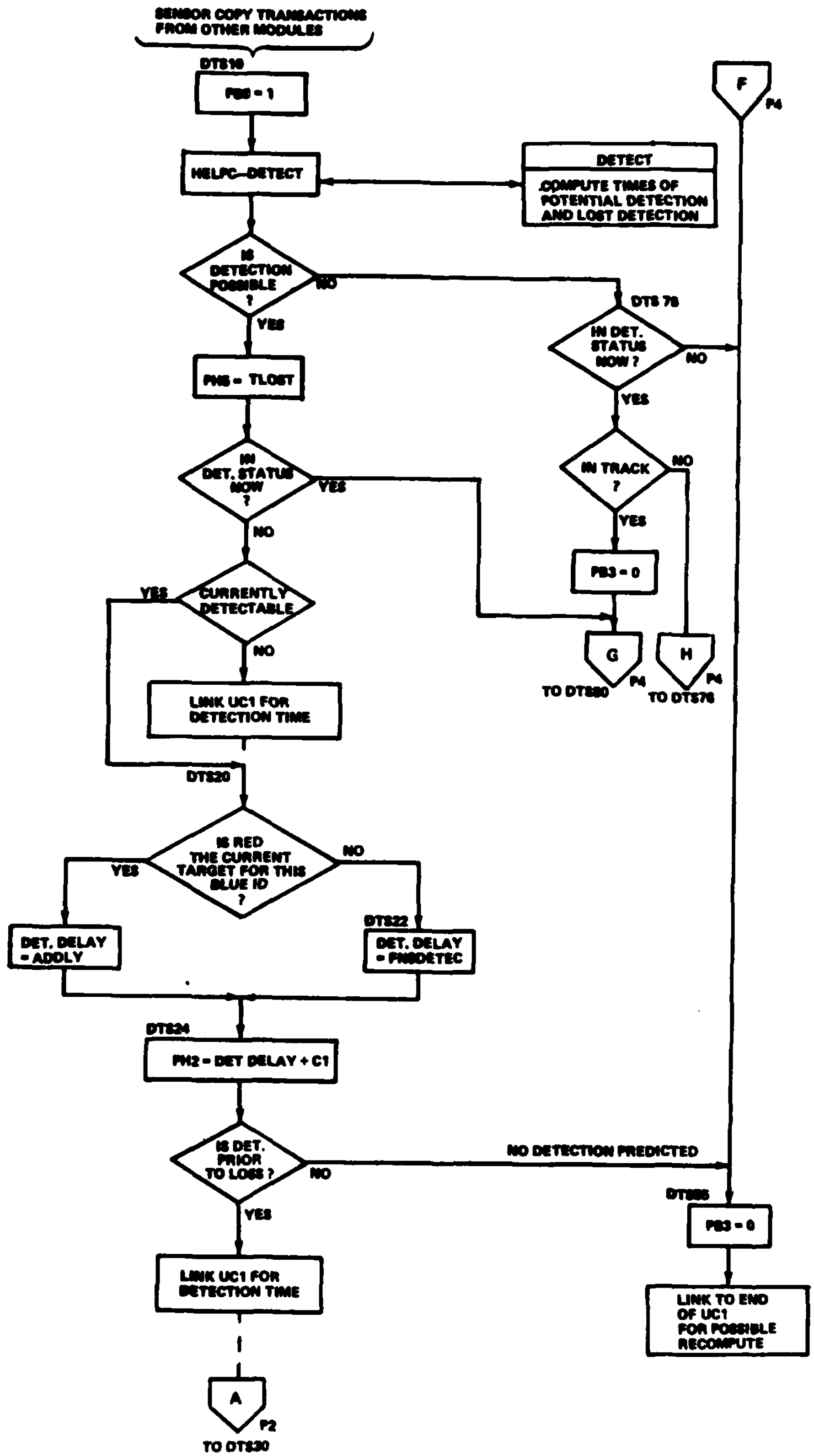


Figure 5-1. Detect and Track Module (GPSS)
(Page 1 of 4)

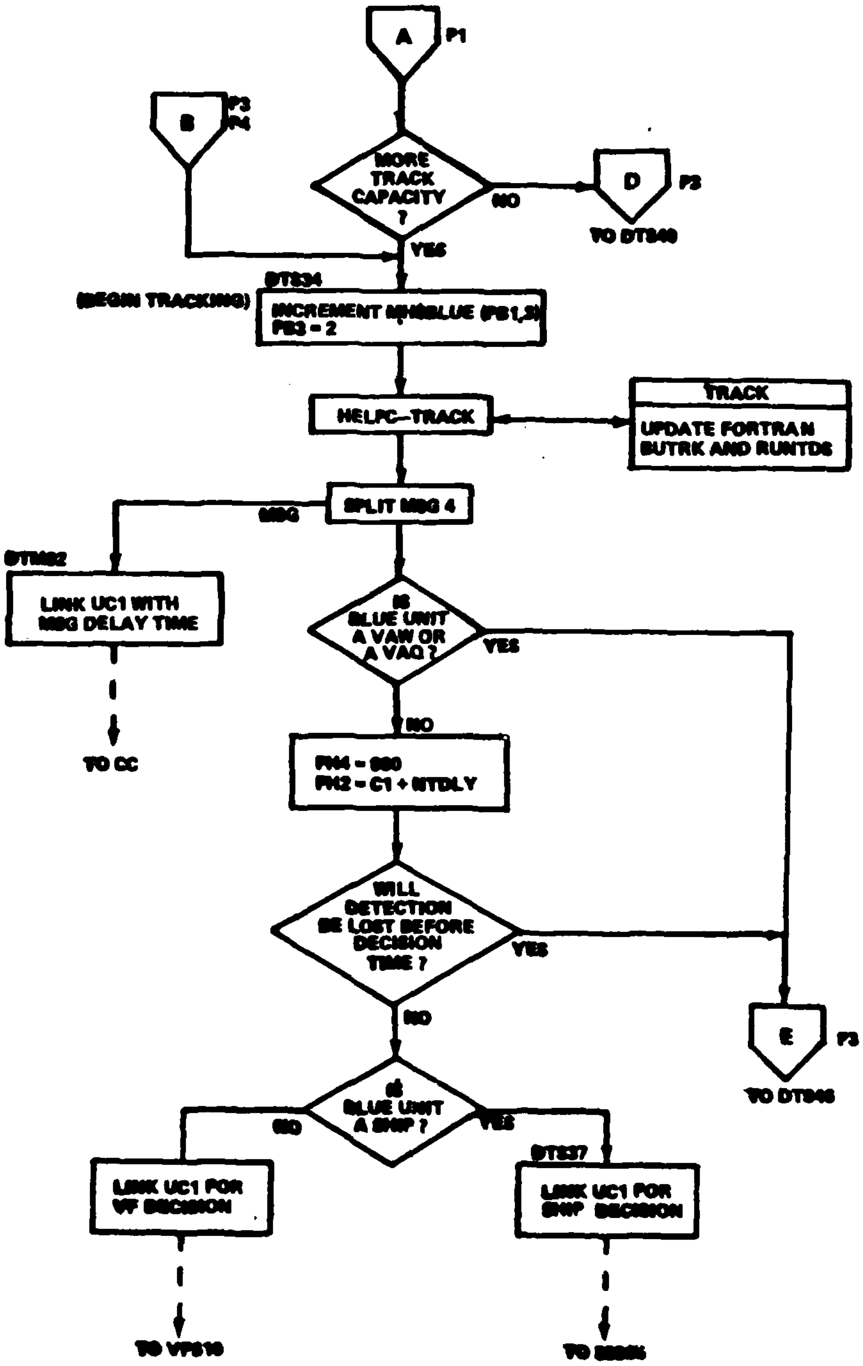


Figure 5-1. Detect and Track Module (GPSS)
(Page 2 of 4)

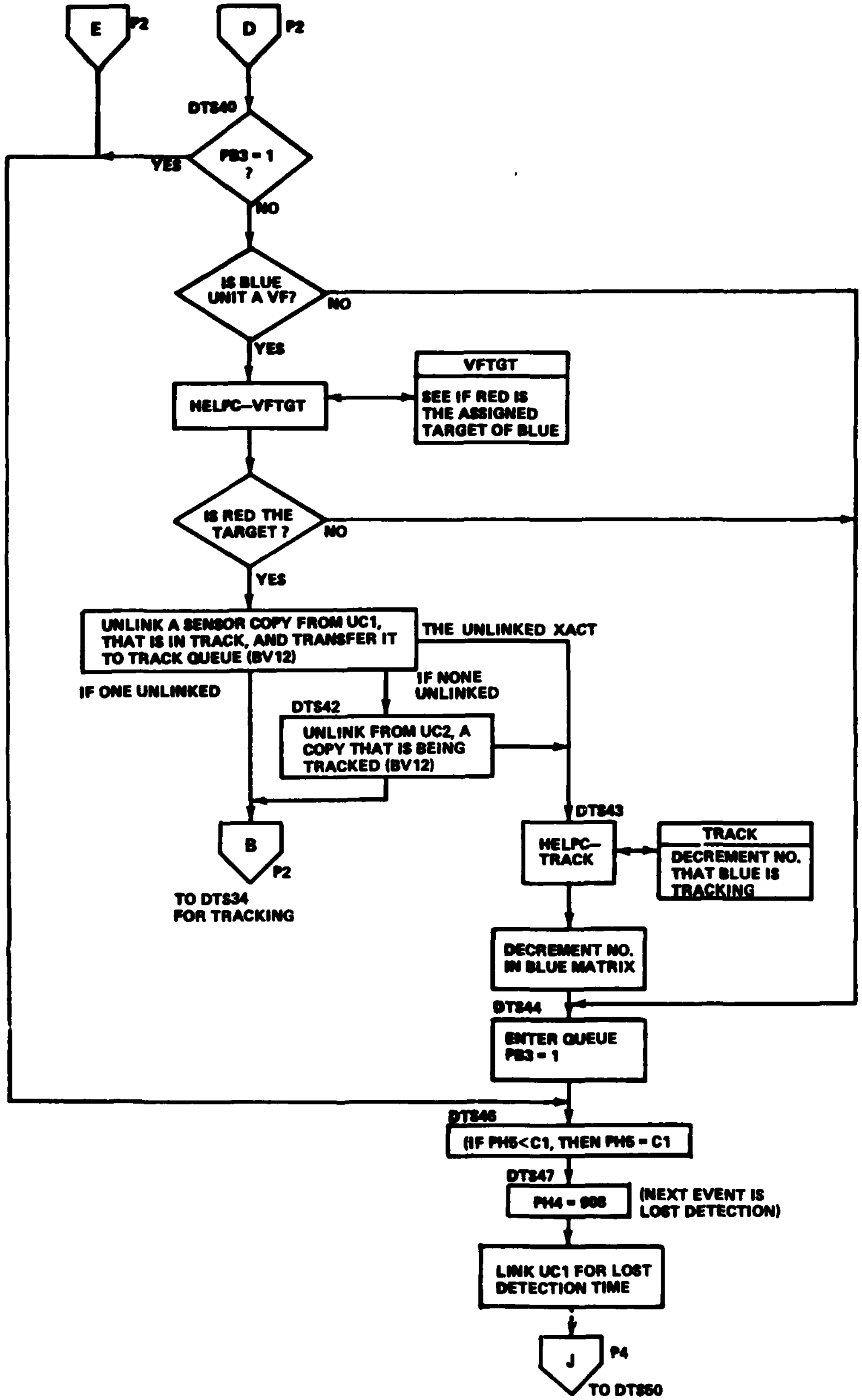


Figure 5-1. Detect and Track Module (GPSS)
(Page 3 of 4)

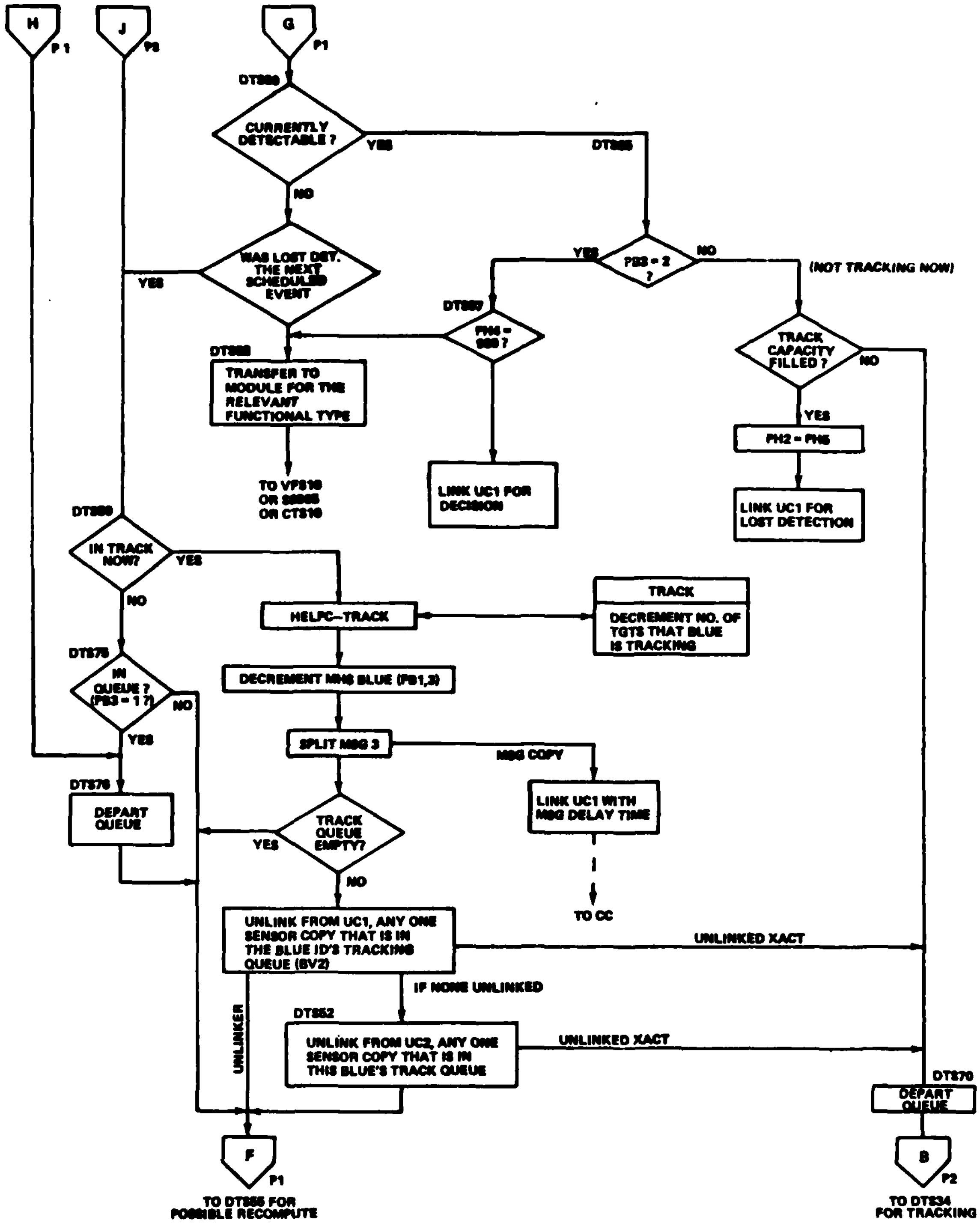


Figure 5-1. Detect and Track Module (GPSS)
(Page 4 of 4)

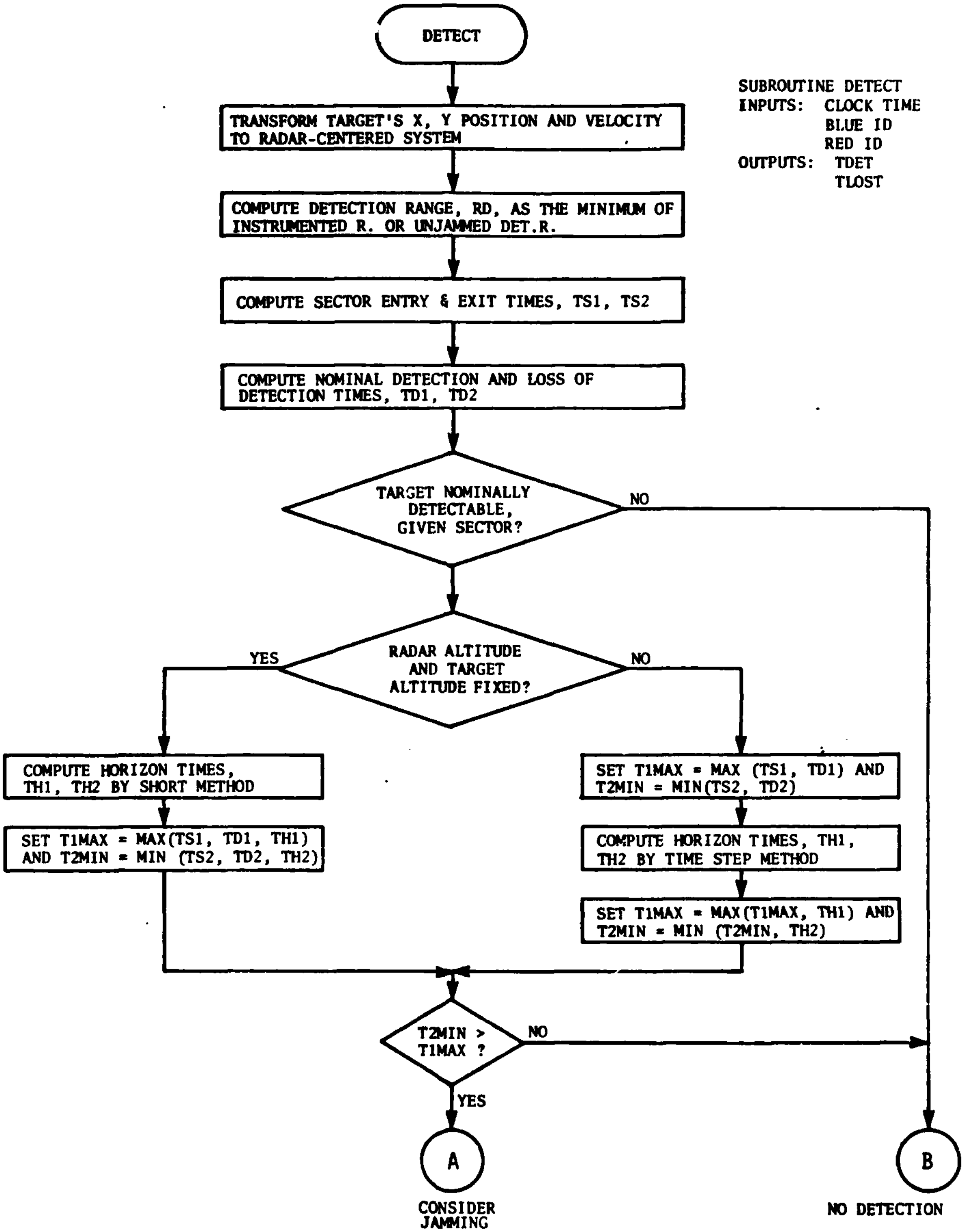


Figure 5-2. Radar Detection Model

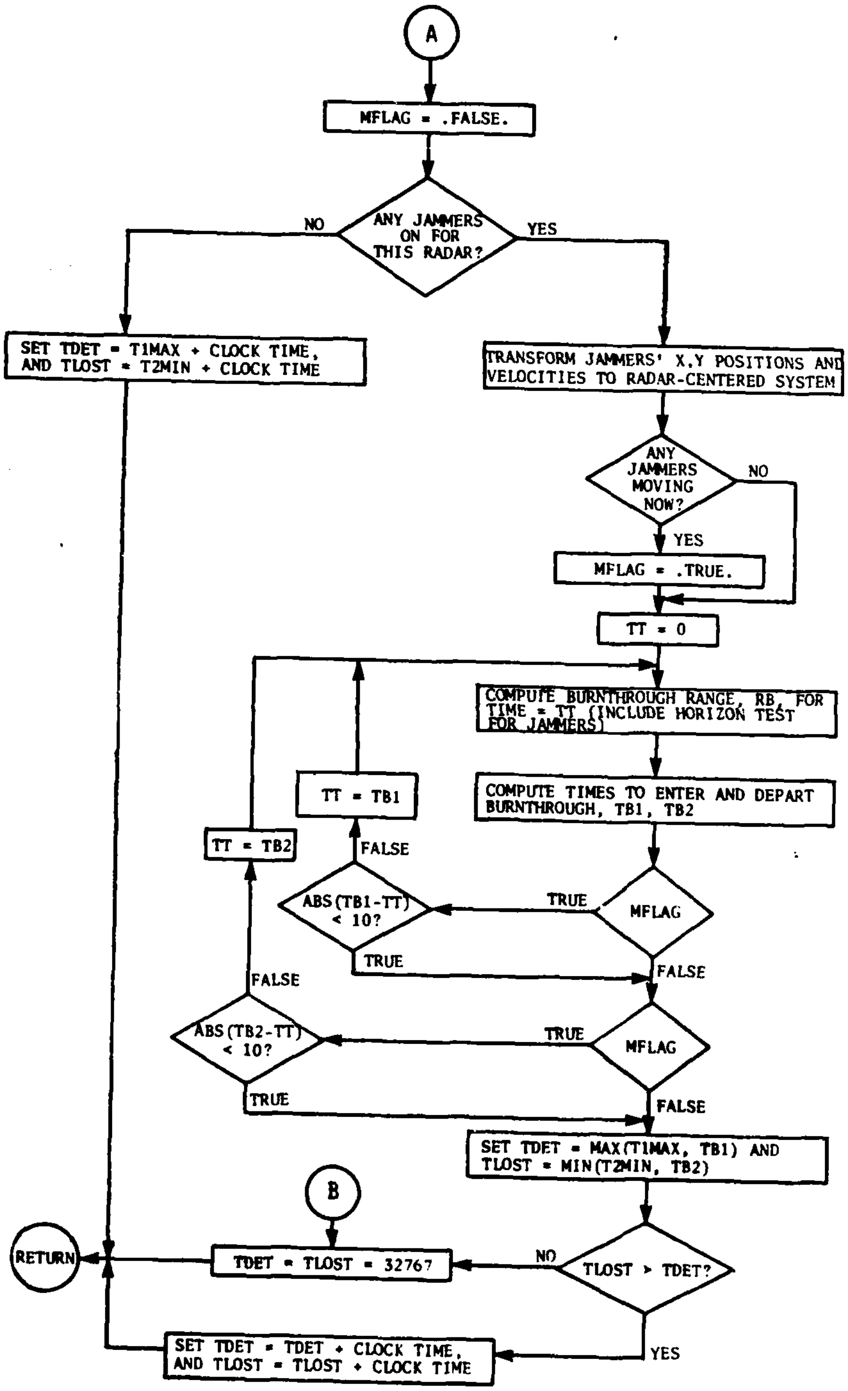


Figure 5-2. Radar Detection Model (cont.)

Time no longer detectable is when the target will violate any of the potential detection conditions if it continues on its present course. The assumptions and computations are similar to those described for time of potential detection.

Time detected is the time an operator would actually make a detection and includes delays for classifying the target. This delay is a function of the track storage contents. The number of tracks in a track storage is assumed to be an indication of how busy the operator is.

The detections are shared with other platforms by sending messages, and by joint membership in the tactical data net. The detections are also used within the detecting platform for decision making.

5.2 JAMMING

Standoff jammers are assumed to be wideband noise jammers. Although they are positioned by the scenario, their location relative to the radar beam pattern is not treated in detail. It is assumed that each jammer is always attenuated by the average sidelobe level of the radar.* This assumption ignores the case where the bearings of the target and the jammer differ by less than one beam width, and there is consequently no sidelobe attenuation. This special case is usually a transient situation, not very significant tactically, and its inclusion would be costly in NADS run time (because DETECT is called thousands of times).

The burnthrough range is computed as

$$R_b^4 = R_a^2 SA\left(\frac{P_1}{D_1^2} + \frac{P_2}{D_2^2} + \cdots \frac{P_n}{D_n^2}\right)^{-1}$$, where

R_b is the Burnthrough Range

R_a is Range Alpha, burnthrough range on a one sq. m. target, self-screen jammed by one watt per MHz.

*If the radar target is also the platform of the jammer, it is recognized as a self-screen case and no sidelobe attenuation is applied.

S is the radar's sidelobe ratio (arithmetic; not dB).

A is the radar cross section of the target.

P_n is the effective radiated power density of jammer number n, in watts per megahertz.

D_n is the distance between radar and jammer n.

The directivity of SOJ's antenna is represented in the data for effective radiated power density in the victim radar's bandwidth. SOJ antenna patterns are usually broad enough to cover all the intended victims. NADS assumes that is the case, and does not require that the scenario include SOJ aiming data.

When jammers operate from stations that are essentially stationary during the ON periods, the burnthrough range is nearly a constant. The target track crosses the fixed circle at readily computed entry and exit times (if it crosses at all). In some tactics, however, jammer motion may be an essential element. In that case a closed form solution is not feasible; the burnthrough time must be known before the correct D values can be obtained to compute the burnthrough range. Consequently, an iterative solution must be provided to handle the moving jammer case. Its use may add substantially to the run time, however. Scenarios should avoid moving jammers during their ON periods unless it is a specific element of the tactics to be examined.

6. COMMAND CENTER MODULE

The Command Center module simulates the air defense decision functions of the battle group and can be thought of as the force Anti-Air Warfare Coordinator. Tactical information is gathered, processed and maintained by the module. Decisions on force employment in the form of Force Orders are issued to individual ships and air control units by this module. Major assumptions regarding the capabilities and limitations of own forces as well as the likely tactics of an aircraft cruise missile attacker are furnished by the user inputs to the model.

Implicit to the model are the military principles of concentration, economy of force and defense in depth. The model will position mobile units in order to bring sufficient fire power to bear to thwart the attack by destroying his forces. The model used economy of force by using the most perishable resources first and by deferring action until action becomes imperative. Multiple engagements are planned for those units which possess such capability. A layered defense of aircraft and missiles is coordinated by the module.

The module responds to the tactical situation with three priorities. The first priority is to meet the current and immediate threat. Hostile aircraft, either unknown or non-fighter types, currently tracked by battle group units are assumed to be a current and immediate threat. The second priority is to position forces to counter the future threat. Position reports furnished by systems and units exogenous to the battle group are assumed to describe the future threat. The module will then attempt to sustain a basic defensive posture. The Air Plan furnished by the user consisting of fixed Combat Air Patrol (CAP) stations and an alert schedule for each aircraft carrier (CV) are assumed to be this defensive posture.

The Command module receives twenty-six (26) distinct message reports as well as one pseudo-message and one special event (No. 888). The message reports are used to update the command information arrays for hostile contacts and battle group units. The peuedo-message (No. 0) is used to issue more orders if the number of orders to be issued exceeds the storage available. This pseudo-message is used only within the Command Module. The Special event is used to schedule most of the decisions concerning force employment.

The Command Module maintains four principal data structures: status arrays, target arrays, group arrays and the IC arrays. The status arrays contain data concerning the status, major subsystems and tactical emloyment of battle group units. The target arrays contain data concerning the hostile elements currently reported by the units of the force. The group arrays contain data about hostile units which have been reported by units outside the battle group. The IC arrays contain data about the current defensive posture of the force including both the fixed positions and the positions assumed in response to the tactical situation. The IC arrays also contain the future planning of the Command Module. The command center module logic is shown in Figure 6-1.

The Command Module maintains the AIRLST, a list of targets which are eligible for assignment to fighter aircraft for prosecution. A SAMLST, a list of targets which are eligible for application of surface-to-air missiles is also maintained.

Table 6-1 summarizes the twenty-six specific messages and the Command Module response. Routing of surveillance messages is shown in Figures 6-2 and 6-3.

The execution of the special (888) event results in the scan of the AIRLST and the SAMLST and the generation of force orders. A waiting period is associated with each order given and the special event reschedued for the earliest expiration of these waiting times. The arrival of a report which could result in the scan of either AIRLST or SAMLST will also reschedule this special event during the current time. This event has low priority so that execution will occur after all messages have been received, thus only one scan of the lists will be made at any clock time.

Table 6-1. Command Messages

MESSAGE NUMBER	MESSAGE INTERPRETATION	PERMISABLE COMMAND MODULE RESPONSE
1	External Surveillance Report	3,4,a,d
3	Lost Track Report	2
4	Tracking Report	2,5
6	Target Accepted Response	1,2,3,4
7	Target Rejected Respone	1,2
8	Hit Target Report	1,2,5,6,z
9	Missed Target (Heads Up) Report	5,6,z
10	Target Track Change	5,c,d,6,z
107	Target is Fighter Report	1,5,6,z
111	Engaging Target Report (Unit Decision)	1,2,5,4,a,d,z
112	Reject Handover to Aircontroller Report	2,g
113	Cannot Comply (CANTCO) with Launch Order	1,2,4
203	Sam Self Assign Request	e,f,6
204	SAM Self Assign Request	e,f,6
206	Self Assign Notice	6
302	Accept Control of Fighter	2
303	Unit Down Report	2
304	Fighter on CAP Station Report	2,d,4,z
306	Change in Air Control Capability Report	2
307	Reject Control of Fighter	2,g
313	SAM Inventory (Conventional)	2,z
314	SAM Inventory (Nuclear)	
316	Fighter on the Way Report	2,g,4
318	Carrier Status Reporting	a,4,z
319	Fighter Enroute to CAP Report	2,d,4,z
320	Return to Base (BINGO)	2,4

LEGEND

1. Update Target Array
2. Update Status Array
3. Update Group Array
4. Update IC Array - Replan Force Disposition
5. Process AIRLST
6. Process SAMLST

a. Issue Launch Order
b. Issue Immediate Launch Order
c. Issue Target Assignment Order
d. Issue go on CAP Order
e. Approve Assignment
f. Disapprove Assignment
g. Issue Controller Assignment
z. Reschedule Command Decision

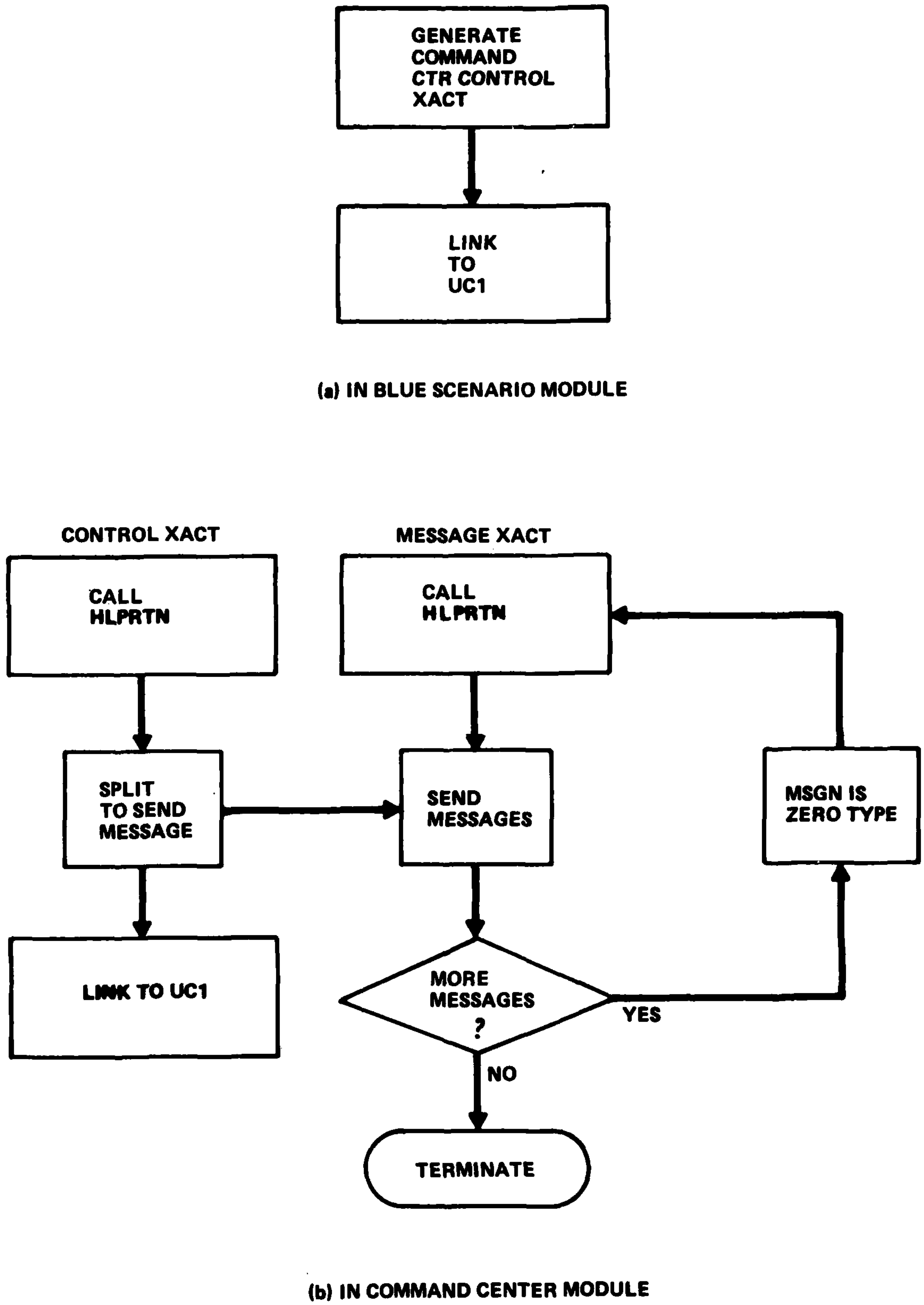


Figure 6-1. Command Center Module, GPSS Elements

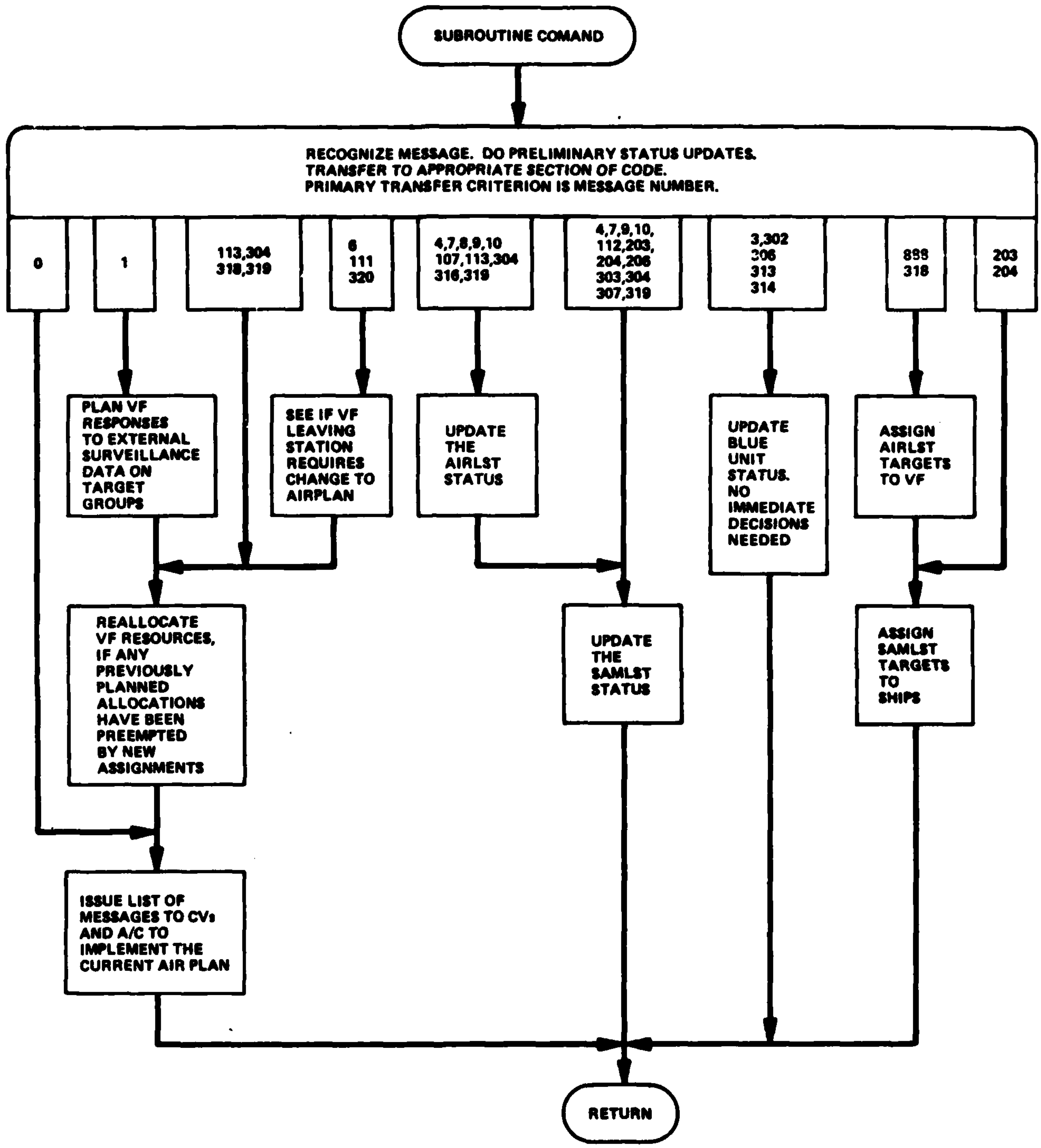


Figure 6-2. Subroutine Comand, Message Routing

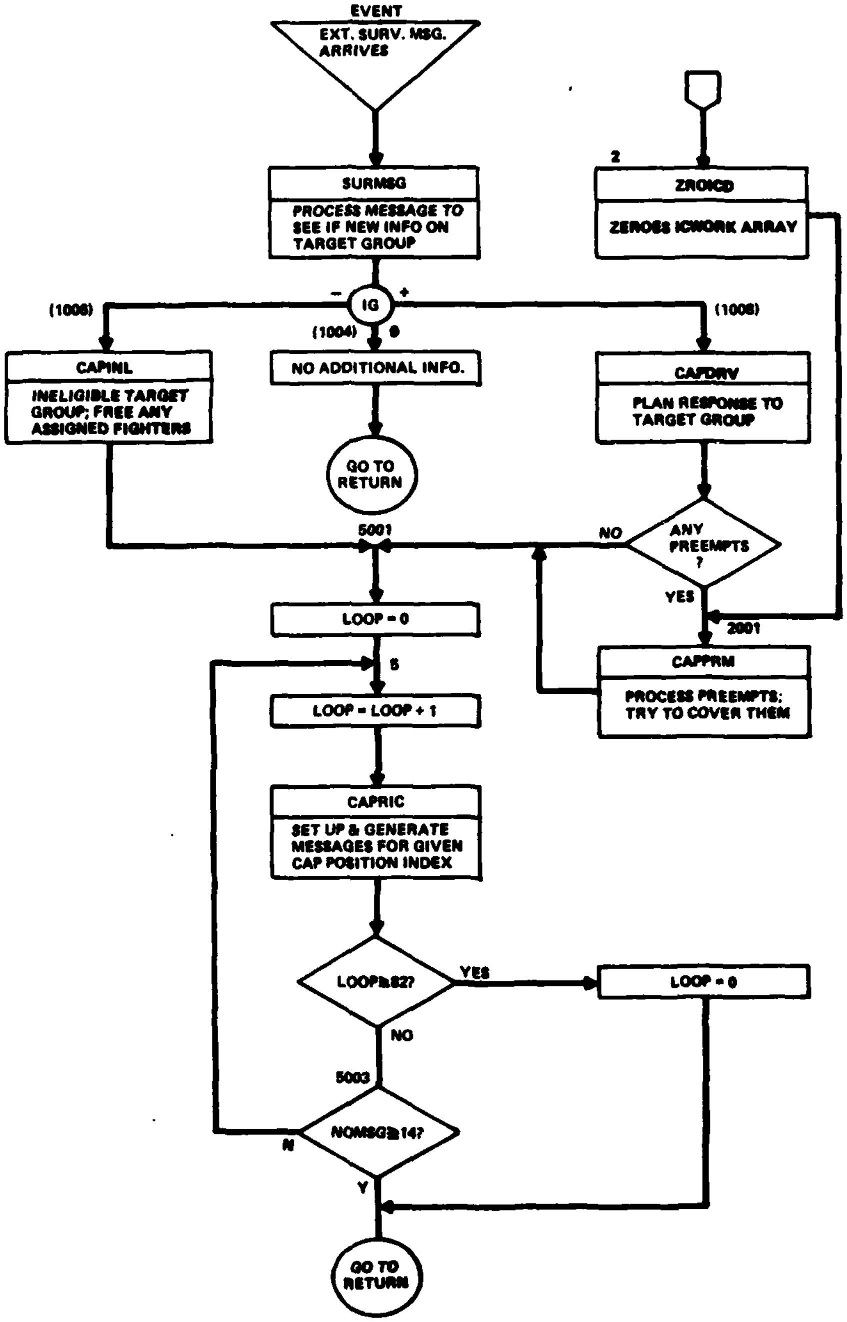


Figure 6-3. Subroutine Comand, Surveillance Messages

6.1 CAP INTERCEPTOR ASSIGNMENT LOGIC

Subroutine TGTCAP performs the function of assigning CAP aircraft to intercept targets. It is called by Subroutine COMAND, the main FORTRAN program of the Command Center Module, when messages are received to report new enemy detections or a CAP becoming available for assignment.

Prior to calling TGTCAP, COMAND will have prepared an updated list of unassigned targets that are suitable for VF intercept. The highest priority targets will have already been tested for possible assignments to DLI's, and where appropriate such assignment will have been made. The targets and assignments are stored in the following variables:

AIRLST(M) - Red IDs of unassigned targets

AIRASN(M) - VF ID of assigned VF (Set negative if secondary assignment, 0 if unassigned) (DLI assignments numbered 1001, 1002,...)

AIRCNT - Number of AIRLST entries to be processed by TGTCAP.

AIRCLK(M) - Time the Command Center will reevaluate the assignment

Two Command Center Status arrays for VFs will be used to indicate in-flight VF availability:

CSVFAV(J) - C^2 idea of VF availability

0 - No target assigned; available for primary assignment

1 - 1 target assigned; available for secondary assignment

2 - 2 targets assigned; currently unavailable

3 - Out of game or insufficient weapons or fuel for another assignment

For these arrays, J is the VD ID, and equals Blue ID - VFBIAS.

Figures 6-4 and 6-5 diagram the assignment logic. The basic concept is to assign CAP that can make the quickest intercepts. Then each

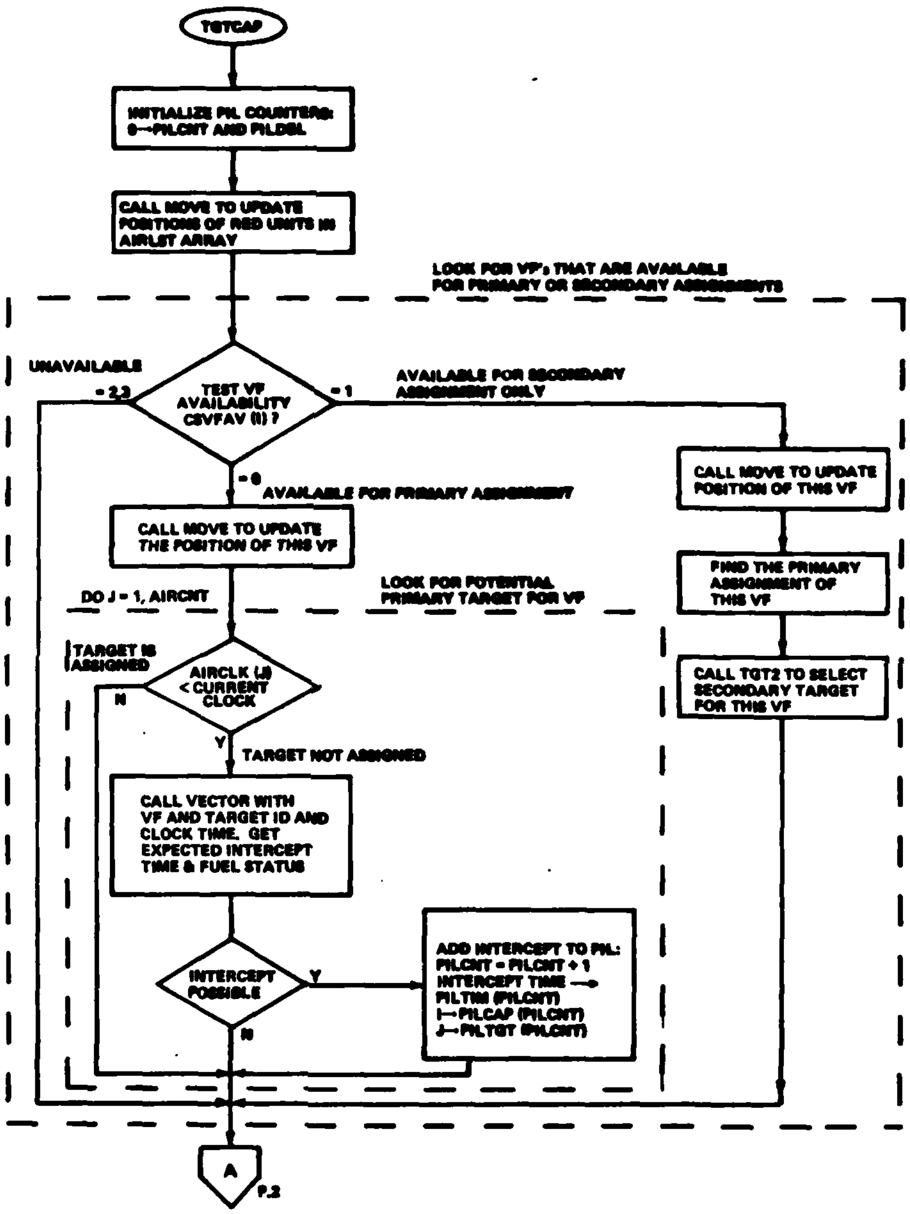


Figure 6-4. Subroutine TGTCAP (called by COMAND in CC Module) (Page 1 of 2)

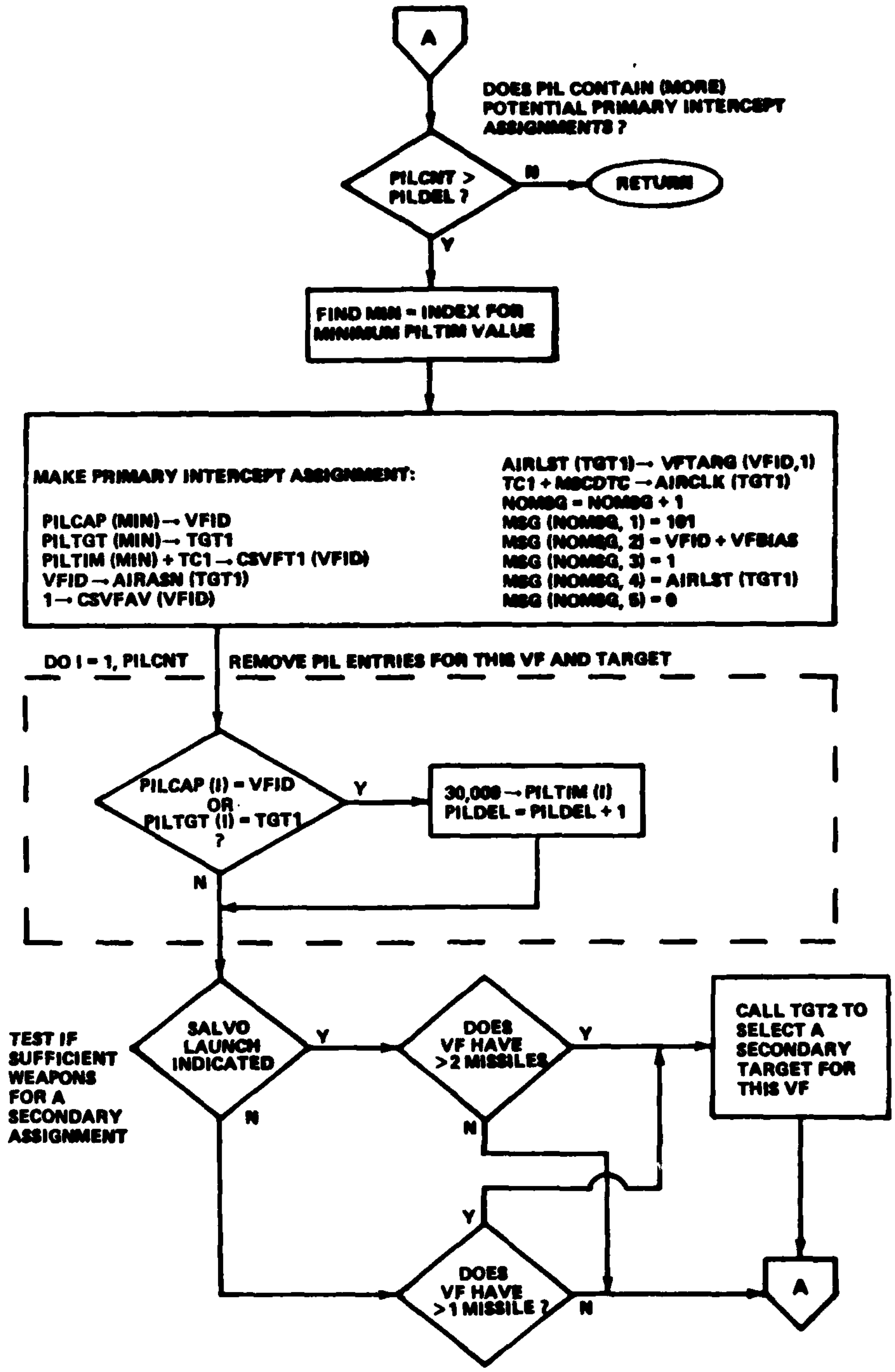


Figure 6-4. Subroutine TGTCAP (called by COMAND in CC Module) (Page 2 of 2)

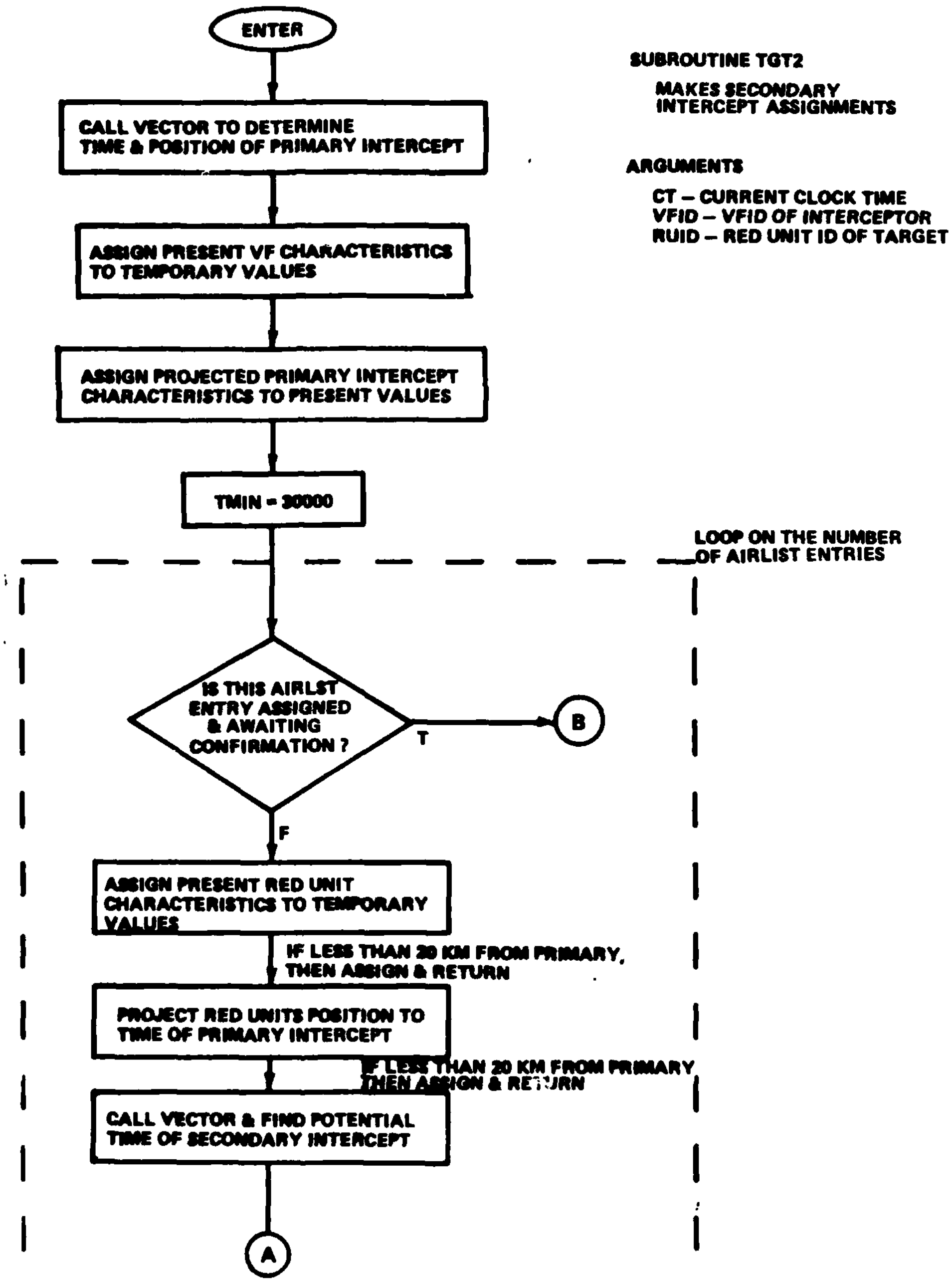


Figure 6-5. Subroutine TGT2 (Page 1 of 2)

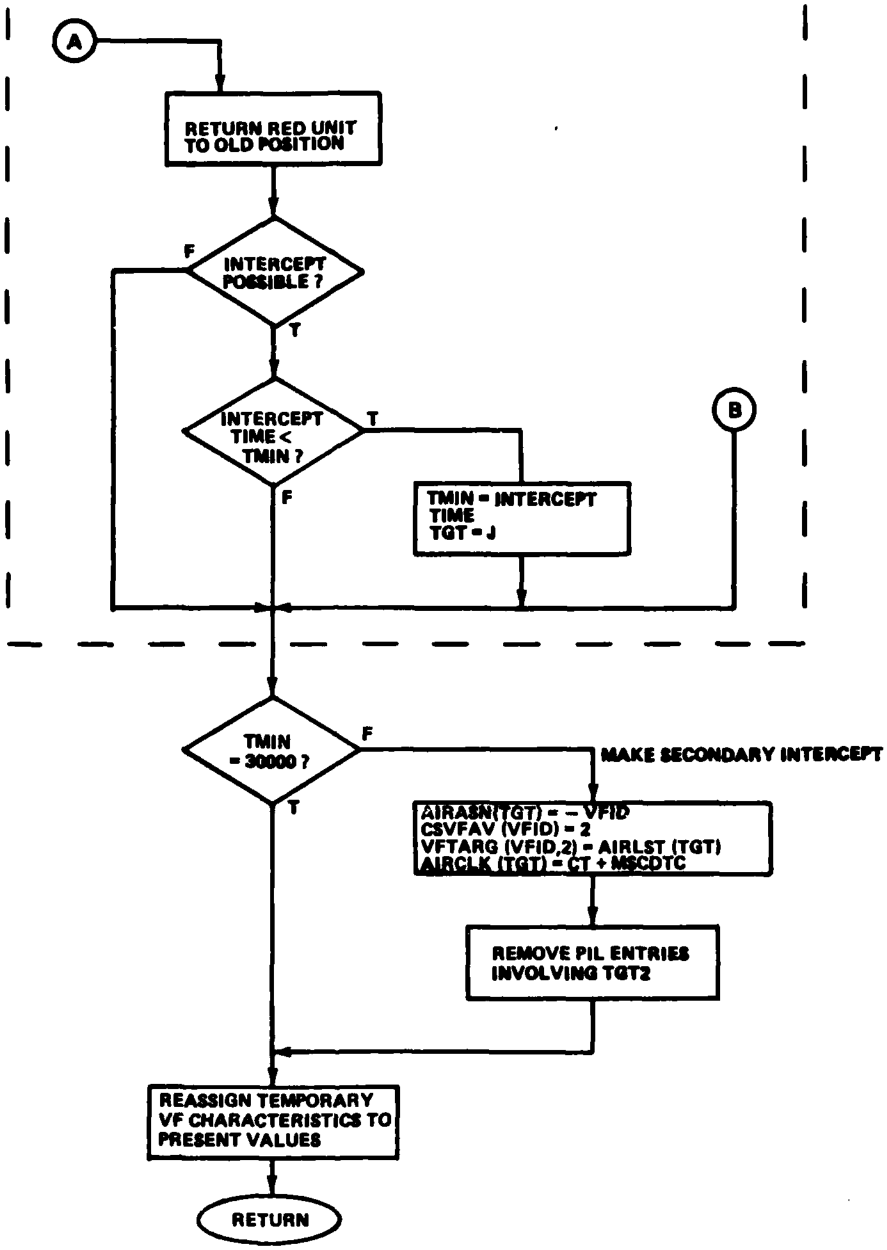


Figure 6-5. Subroutine TGT2 (Page 2 of 2)

interceptor is tested to determine if it can intercept a second target. The concept is illustrated in Figure 6-6. All second target assignments will be tentative and will be reevaluated when a primary target assignment has been completed. Note that second target assignments to VF that are already on an intercept will be made prior to making primary assignments to unassigned CAP. This will help to spread out the VF and make more efficient use of them.

The subroutine logic includes the building of a Primary Intercept List (PIL) which is built by computing the intercept time of each CAP for all targets. All potential intercepts are entered on the PIL, which is composed of the following data arrays and variables:

PILCAP(N) - VF ID of CAP

PILTGT(N) - AIRLST index of target

PILTIM(N) - Time of intercept for VF-target pair

PILCNT - Count of entries in PIL

PILDEL - Count of deletions from PIL

When all target-CAP pairs are computed, the PIL is examined to select the pair with the minimum intercept time. This target is assigned to the paired CAP via the AIRASN array. The CAP's availability is set to 1 in the CSVAV array and the intercept time is stored in CSVFT1. All PIL entries for either the CAP or the target are deleted from the PIL.

If the CAP has sufficient weapons for a tentative second assignment, Subroutine TGT2 is called to make the assignment. When processing of the CAP is complete, the PIL is reexamined to determine the next smallest intercept time remaining on the list.

Subroutine TGT2 attempts to select a secondary target for an interceptor with a primary assignment. This is done by first tentatively moving the interceptor and potential target to the time of the primary intercept. The status arrays are not updated because this is a tentative move, and it usually will be made a number of times for tests. In seeking

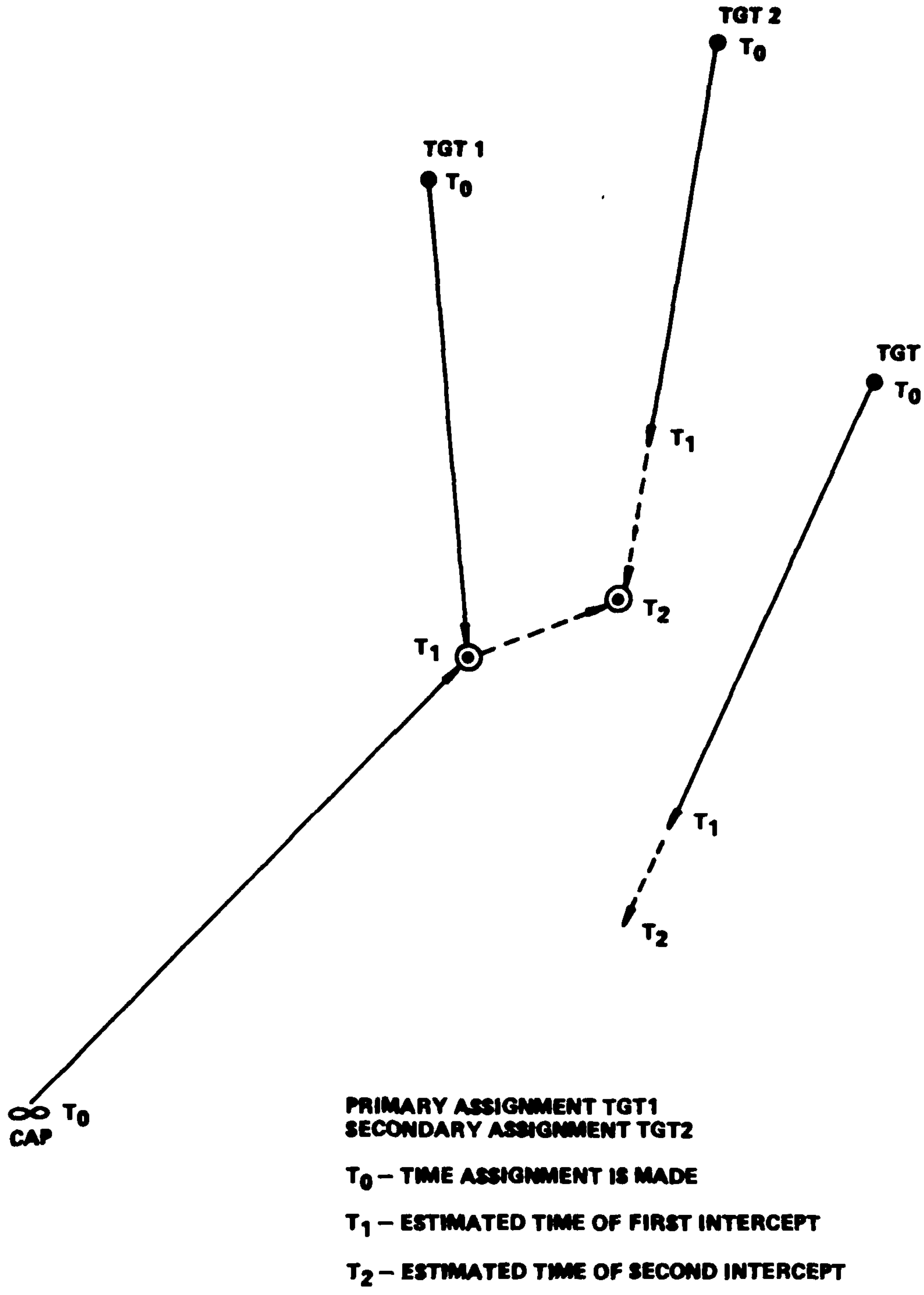


Figure 6-6. Primary and Secondary Intercept Assignments

a secondary target, every unassigned target in AIRLST will be tested to determine which one can be intercepted in the shortest time. The selected target (if any) is assigned via the AIRASN array. The VFID is set negative to differentiate between primary and secondary assignments. All entries for this target are removed from the PIL.

6.2 SAM SHIP ASSIGNMENT LOGIC (TGTSAM)

The SAM Ship assignment logic is entered only if the user has selected Command Center coordination doctrine (CDOCT=1). The Command Center prioritizes targets according to their distance from the defended point and assigns each target to the SAM ship that can make the earliest intercept. The assignments take into consideration how busy each ship is.

A diagram of TGTSAM is shown in Figure 6-7. New detections or Red changes cause the target list (SAMLST) to be updated. Some of the important parameters to be considered are illustrated in Figure 6-8. A target that will reach the defended point first will receive the highest priority. Highest priority targets are assigned first.

The Command Center maintains a list of ships (SHPLST) to which it may assign targets. The Command Center's perception of the number of targets a ship can handle is based on the number of Fire Control Channels (CHAN) each ship has and whether the channels are occupied (CHOC). The Command Center will not make any assignments to a ship that has all channels occupied. The target's projected closest point of approach (CPA) to each ship is compared with the ship's SAM cross-range capability, and targets with CPAs beyond the cross-range are not considered for assignment. The next test is based on when the target reaches the SAM envelope. The SAM envelope that the ship sees is used, but a simplified envelope could be substituted. All ships are checked, and the one whose envelope is reached first by the target is given the assignment. Care is taken not to attempt to reassign a target immediately to ships that have already rejected or missed that target.

The Command Center will give a self assign "NO GO" to a ship self assign request if the target in question has already been assigned to

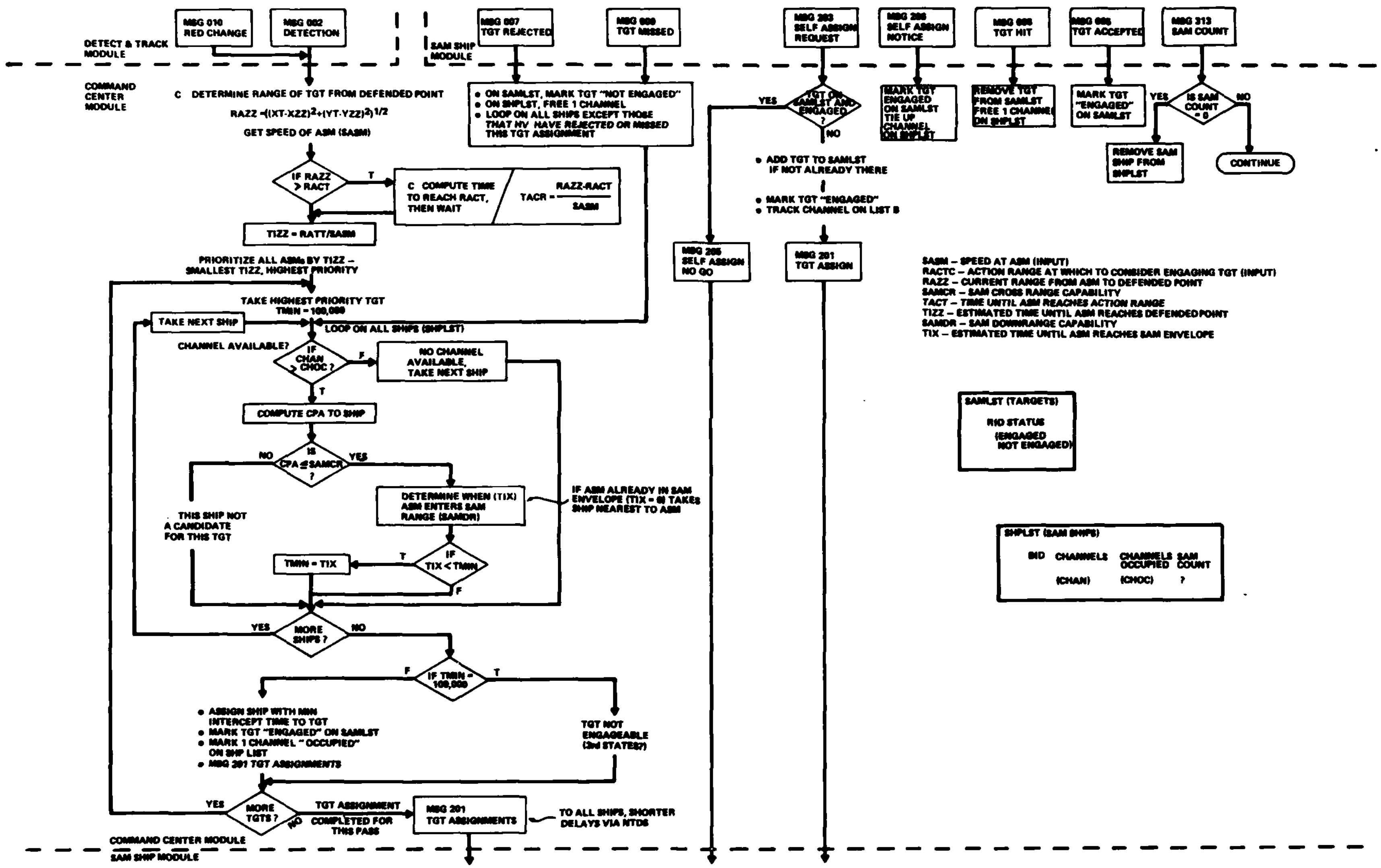


Figure 6-7. Command Center SAM Assignment Functions (TGTSAM)

another ship. A self assign notice message (sent by a ship when it is firing on a target threatening ownship) is not evaluated for a potential "NO GO" response. "Target Hit" messages cause targets to be removed or appropriately marked on the SAMLST. "Target accepted" messages are received by the Command Center, but no actions are considered at this time. "SAM count" messages may not be necessary since all the Command Center needs to know is when a ship is out of SAMs, and this information is available by other means

6.3 TACTICAL RESPONSE TO EXOGENOUS SURVEILLANCE REPORTS

The NADS has provision for entering formatted reports of RED units containing positional data as well as composition information. These reports are processed sequentially by the command center in order of delivery time. This processing causes the repositioning of BLUE movable units, fighter aircraft.

The logic for changing the disposition of the fighters is the result of a few basic principles. The fighter should be positioned in such a way that intercept is feasible when the RED unit maneuvers to close the force. Intercept should be accomplished before the RED unit's distance from the defended point is greater than some keep out range.

Fighters should maneuver while repositioning at an efficient speed, maximum range performance. Any force adjustments should be timed for the latest time possible.

It is assumed that the reporting interval for the systems, combined with the delay from observation to receipt of a finished report, is greater than the expected time between target maneuvers. This assumption means that usual methods for predicting future positions give poor results. The NADS assumes a maximum speed for the RED threat and attempts to interpose the fighter between the threat and the defended point.

Given a position report on a potential RED threat an envelope of fighter positions, which can result in intercept before the keep out range, can be constructed. The most stressing target maneuver is, of course, the direct flight to the defended point. Figure 6-9a is the intercept envelope with the fighter speed greater than the threat speed.

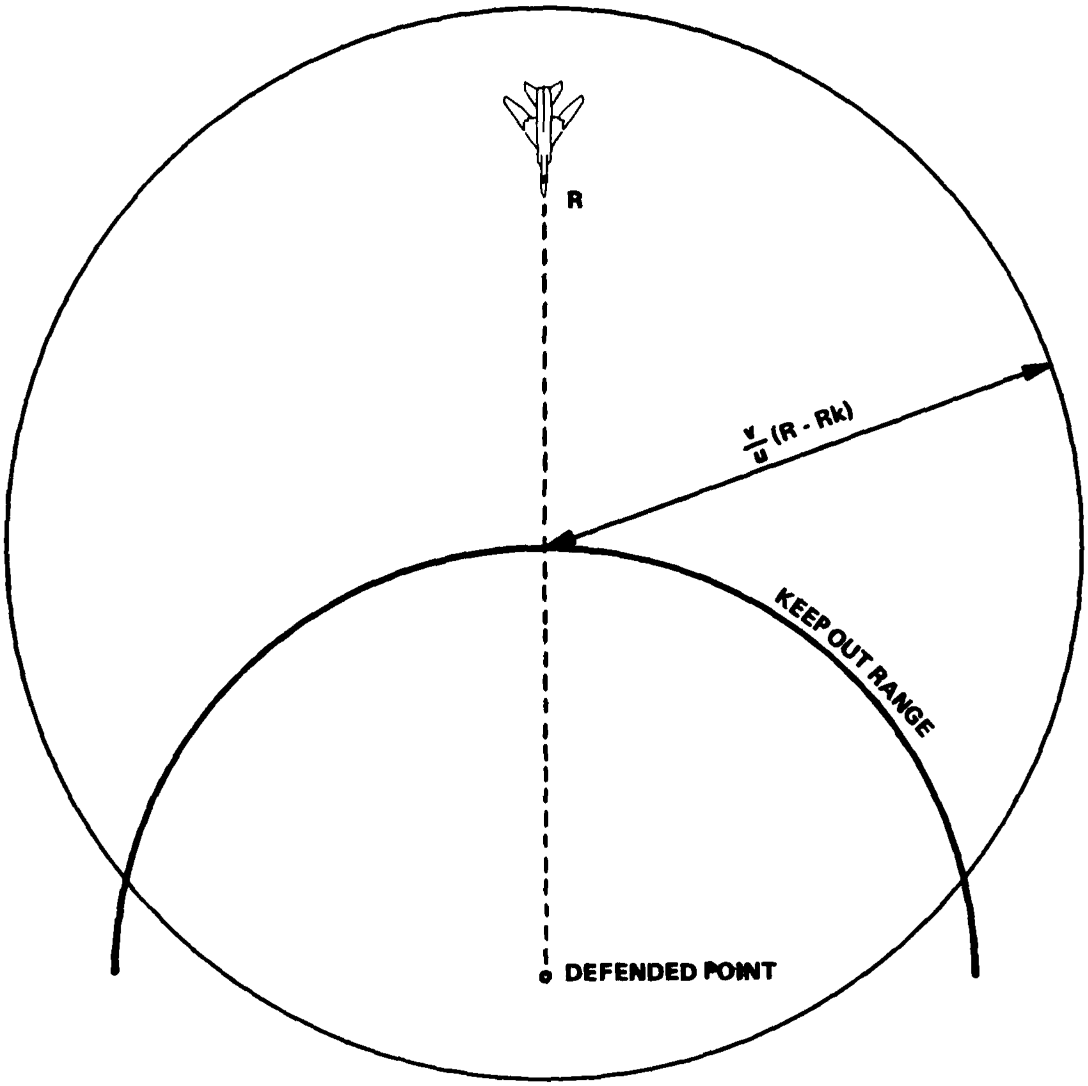


Figure 6-9a. Intercept Envelope v > u

Figure 6-9b is the intercept envelope with the threat speed greater than the fighter speed. This intercept envelope is for a simple collision. It can be argued that any growth in the envelope due to weapon range compensates for factors such as: initial fighter heading; variability in the track; multiple shots; response delays; and initial target positional uncertainty.

As time passes, the size and shape of the intercept envelope also changes. The target is located in a circle with radius u Δt centered on the last observation where u is the maximum speed of the threat and Δt is the elapsed time since the observation. The most stressing target maneuvers are the direct flight to the defended point and the flights which maximize the change in bearing from the defended point. These maneuvers as well as the resulting intercept envelopes are shown in Figure 6-10. The intersection of the individuals envelopes is the envelope of intercept independent of target maneuver. This envelope is cross-hatched in Figure 6-10. As long as the fighter is in this area, the RED threat can be intercepted before the keep out range.

As time passes, the envelope of intercept becomes smaller. It can be shown that the rate of collapse is the speed of the fighter, v. If the fighter waits until the boundary passes, a position can be maintained within the envelope. This policy is consistent with our basic principles.

The envelope of intercept collapse continues to a single point at some time. This point is on the keep out range at the bearing of the last target position report.

Since, in general, the fighter will be within the keep out range in NADS, the complication of the cuspoid shape can be disregarded and a rather basic rule stated for the repositioning of the fighter. The fighter should proceed to the keep out range and the bearing of the last target position report to arrive at time $(R-R_k)/u$ after the observation.

Provision for managing the defensive posture using this basic rule was implemented in NADS using fifteen subroutines. Subroutine CAPAIR reviews airborne fighters without an assignment for use in response to an observed target or to maintain a fixed defensive posture. A flow chart for this routine is Figure 6-11. Subroutine CAPCV reviews unassigned

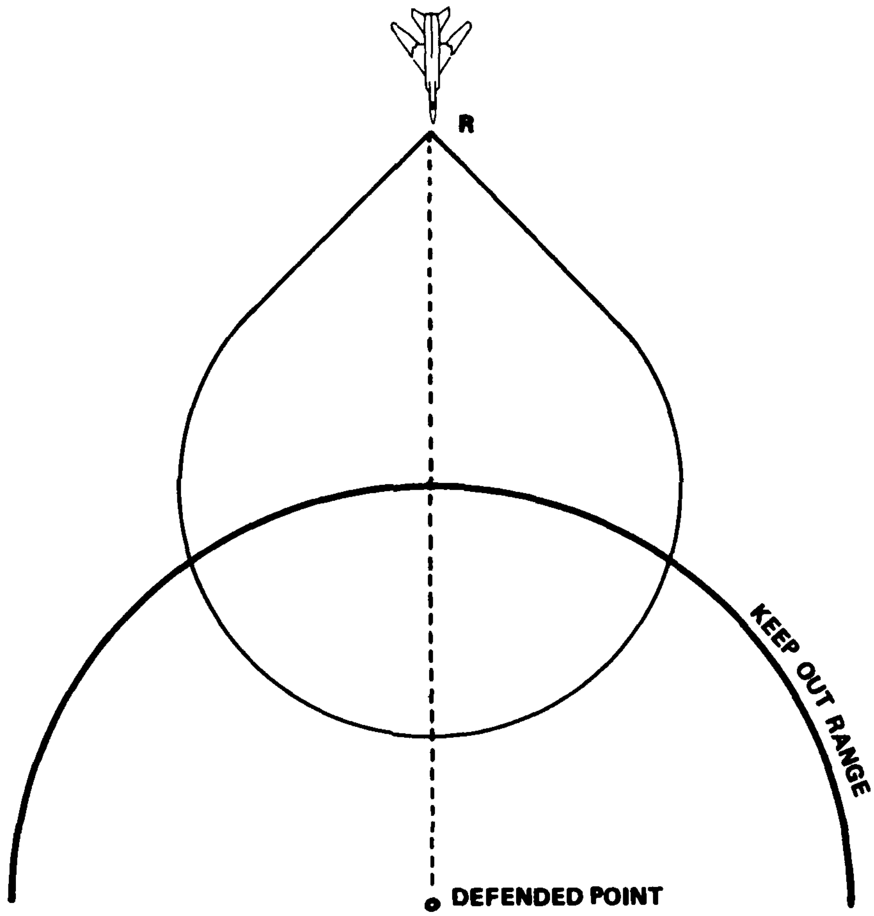


Figure 6-9b. Intercept Envelope $v \leq u$

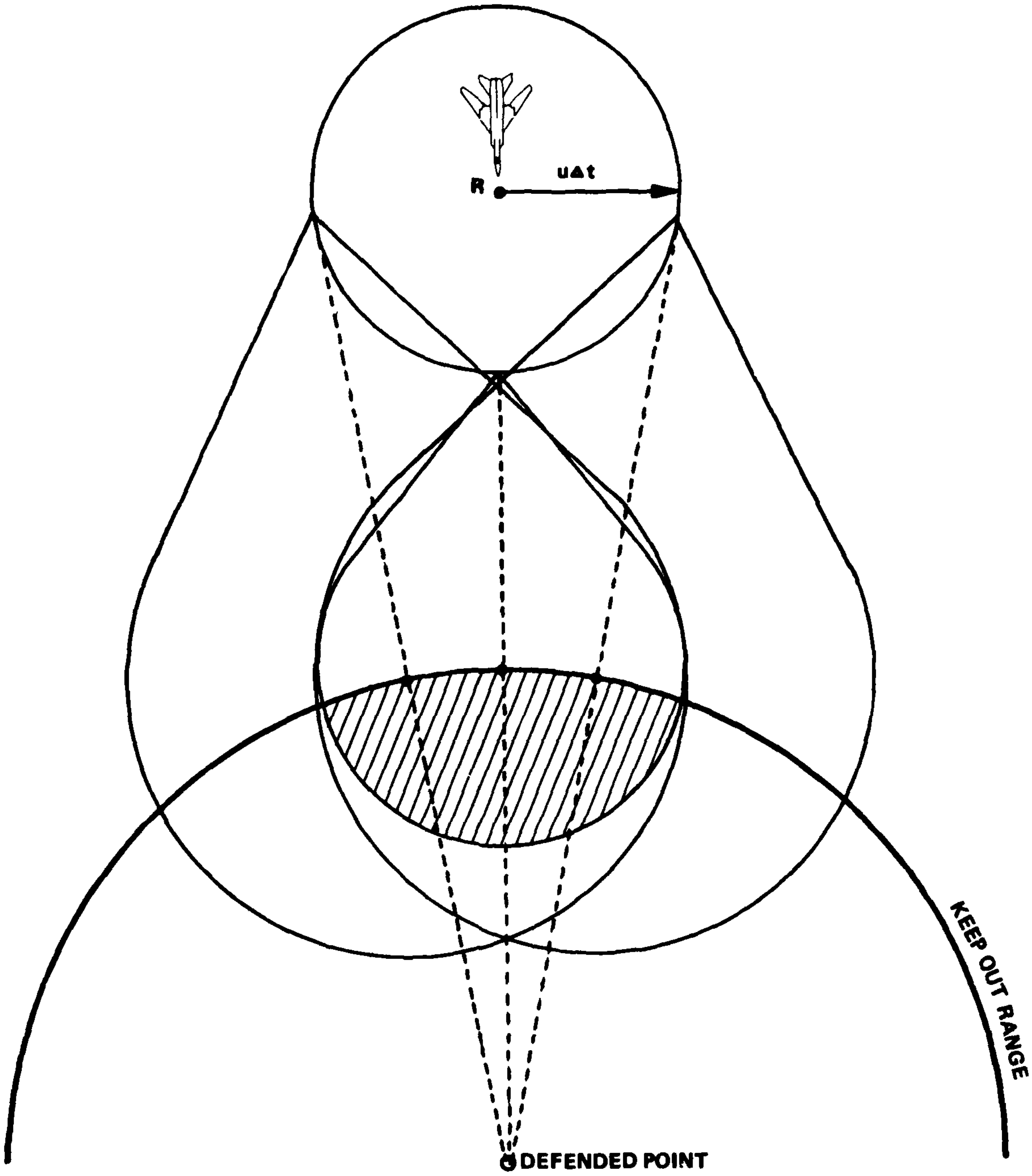


Figure 6-10. Intercept Envelope at Time ΔT

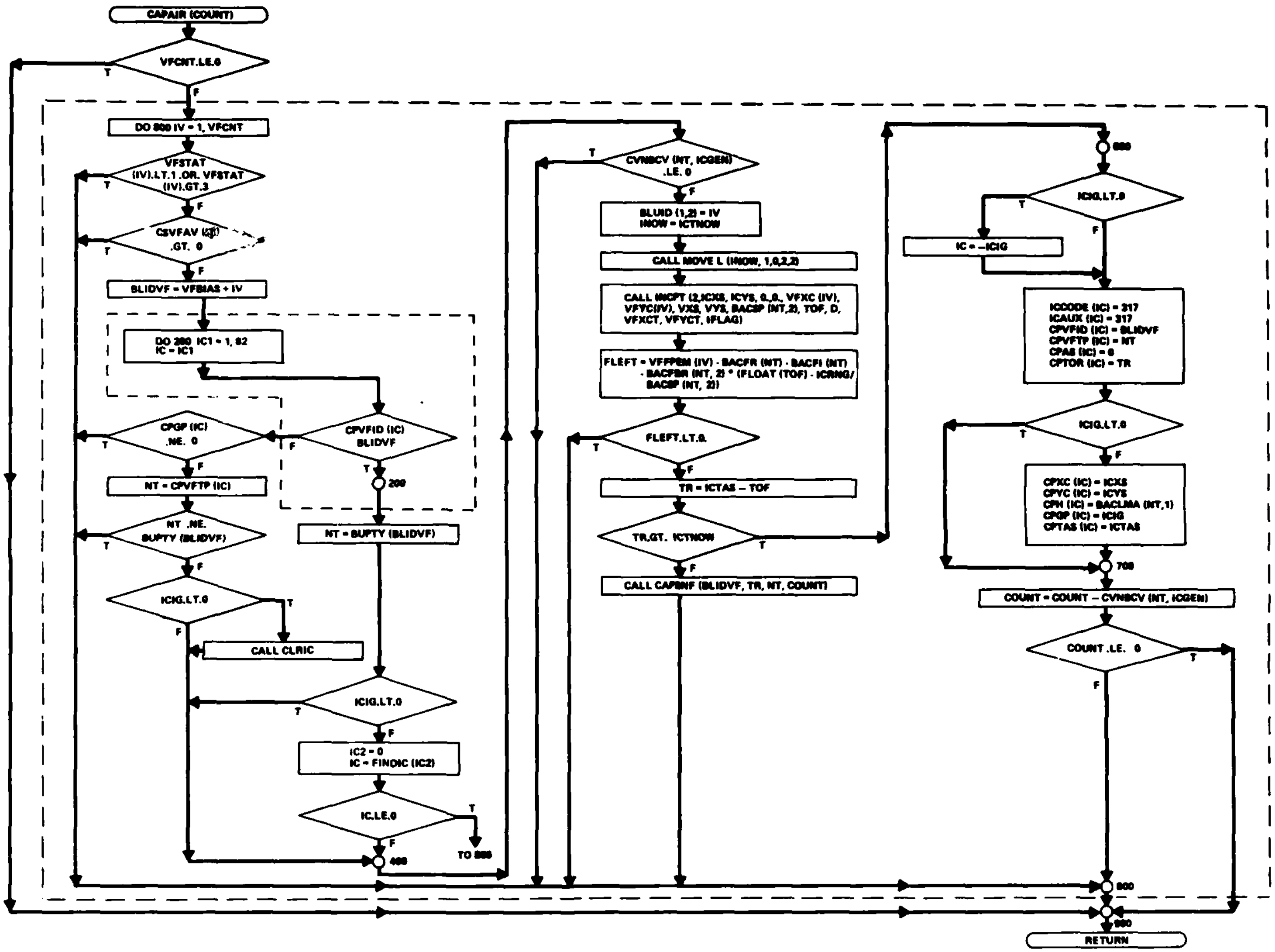


Figure 6-11. Subroutine CAPAIR

(rev. 10/15/82)

fighters on aircraft carriers for use in response to an observed target group or to maintain a fixed defensive posture. A flow chart for this routine is Figure 6-12. Subroutine CAPDRV determines use of airborne and on-deck fighters in response to an observed target group or in filling fixed stations. The order of review in response to a target group is current plan adjustment, assignment of idle airborne fighters, assignment of idle shipborne fighters, preemption of shipborne assignments, preemption of airborne fighter assignments, and unassigned fighters which may not respond in time. A flow chart for this routine is Figure 6-13. Subroutine CAPGRP reviews and adjusts as necessary the fighter assignments that constitute the current plan to cover an observed target group. A flow chart for this subroutine is Figure 6-14. It should be noted that targets seldom make the worst case maneuver which the plan is designed to cover. As more information is received, the most likely result is to move the time of response, the time the fighter must begin maneuvering, to a later time.

Subroutine CAPINL frees any fighters which are assigned to a target group which has been reclassified as RED fighters. A flow chart for this routine is Figure 6-15. Subroutine CAPPAI is used to consider preempting airborne fighters to respond to an observed group. The priority among activities is: first, prosecute RED threats being reported by the sensors of the battle group; second, position forces to block RED threats reported in exogenous reports; and last, maintain a fixed defensive posture. Within the positioning of forces to meet the threat reported by surveillance systems, priority is based on time of response, with soonest response time being most important. A flow chart for this routine is Figure 6-16.

Subroutine CAPCV considers preempting shipborne fighters for use in response to an observed target group or in filling a fixed station. A flow chart for this subroutine is Figure 6-17.

Subroutine CAPPRM reviews the list of stations from which fighters have been preempted and continues processing in order to cover the threat

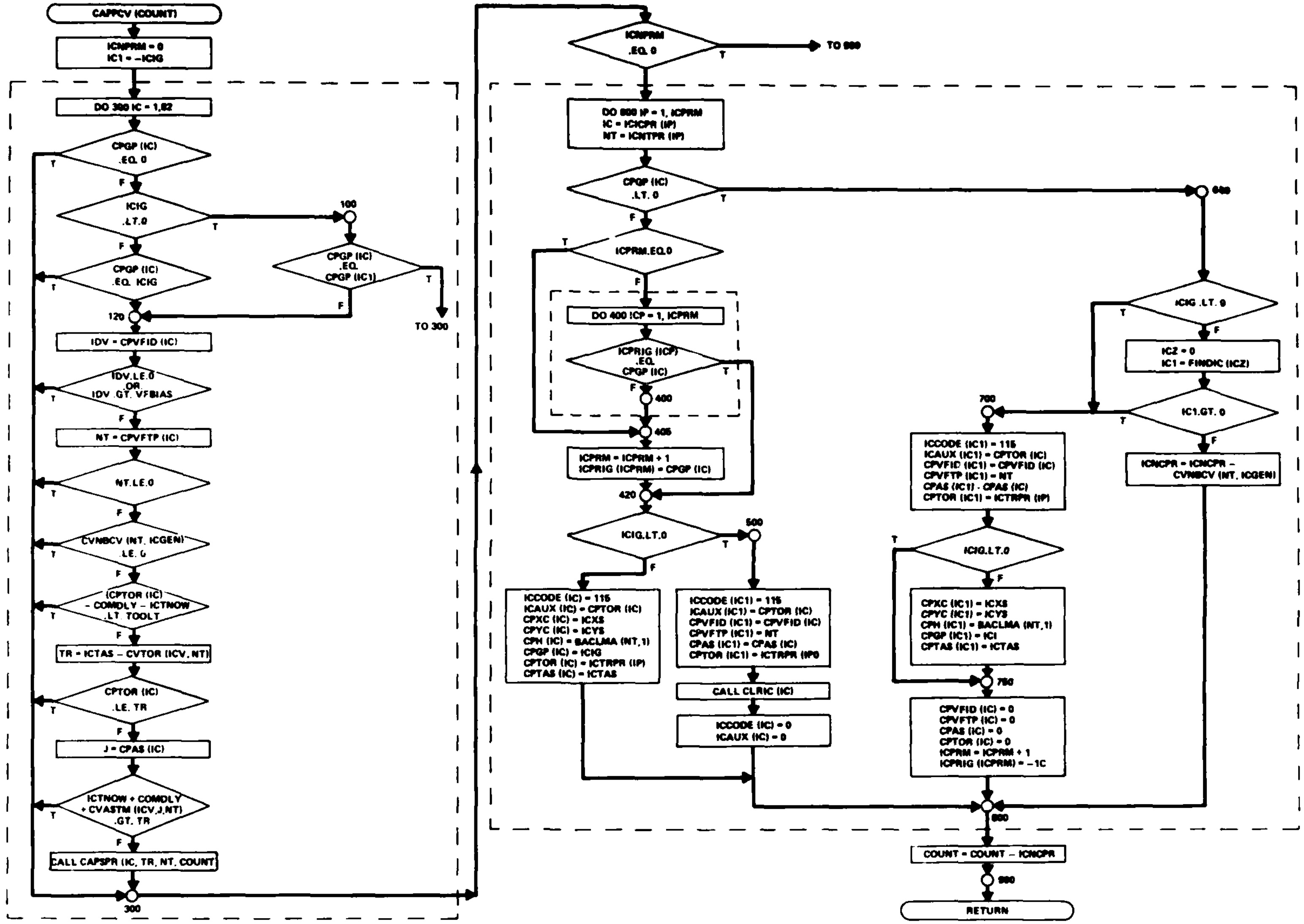


Figure 6-12. Subroutine CAPPCV

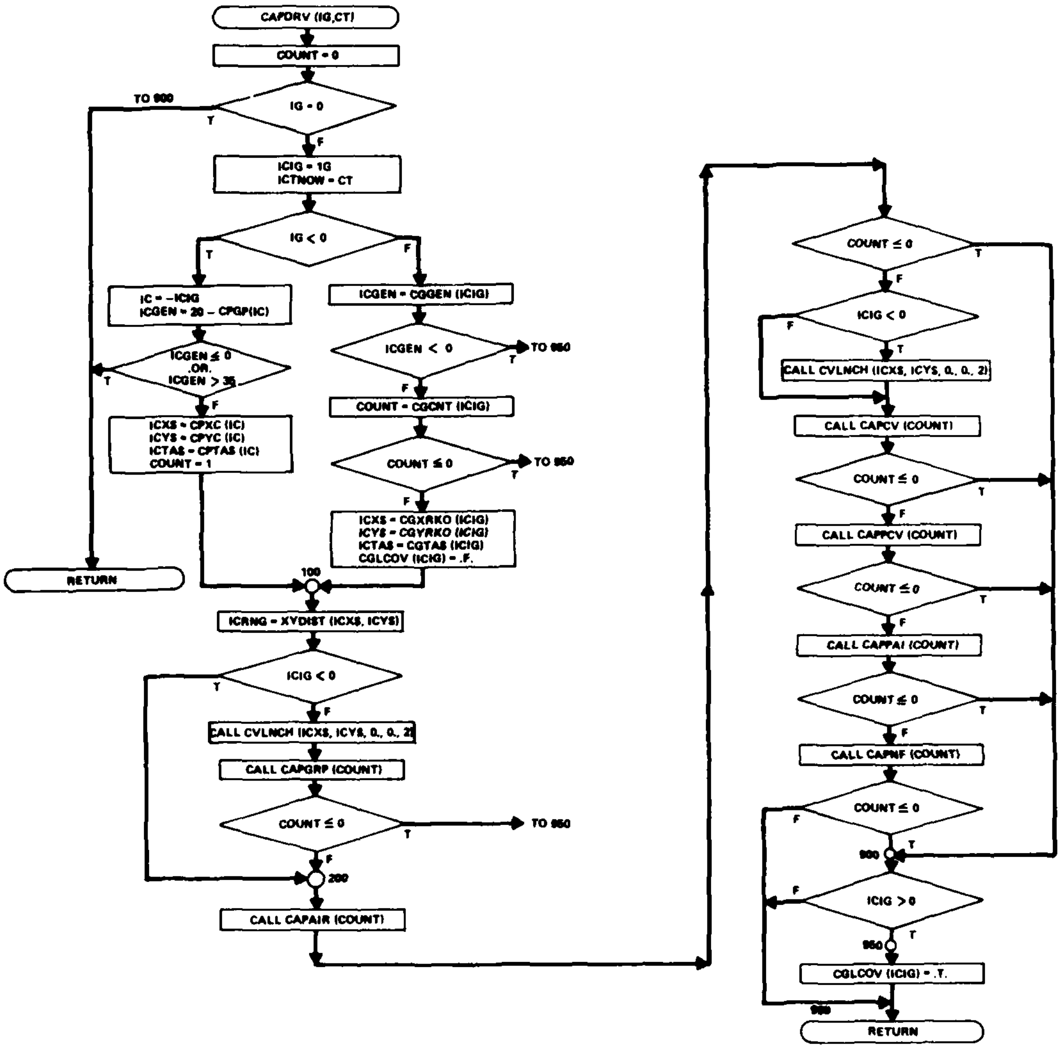


Figure 6-13. Subroutine CAPDRV

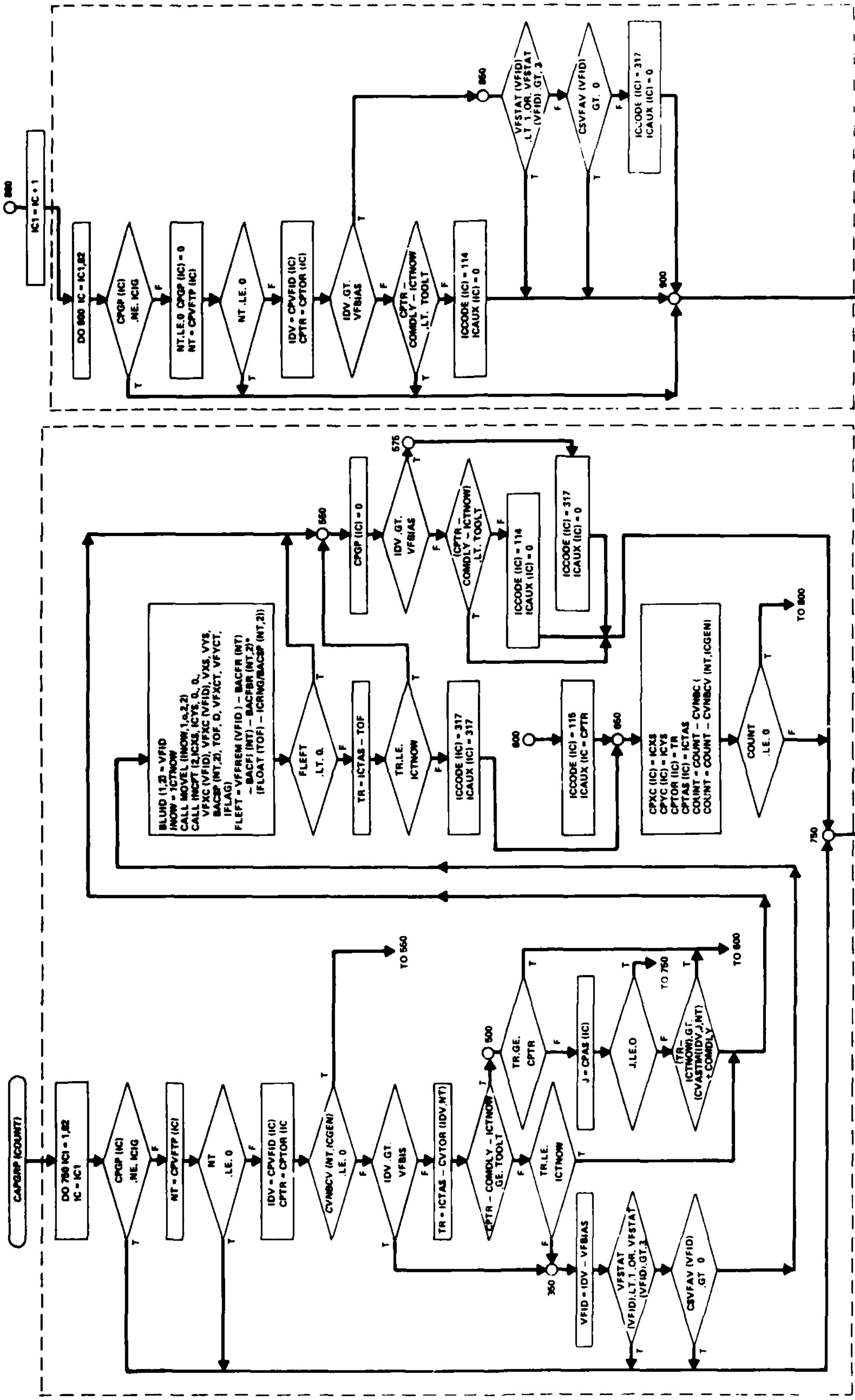


Figure 6-14. Subroutine CAPGRP

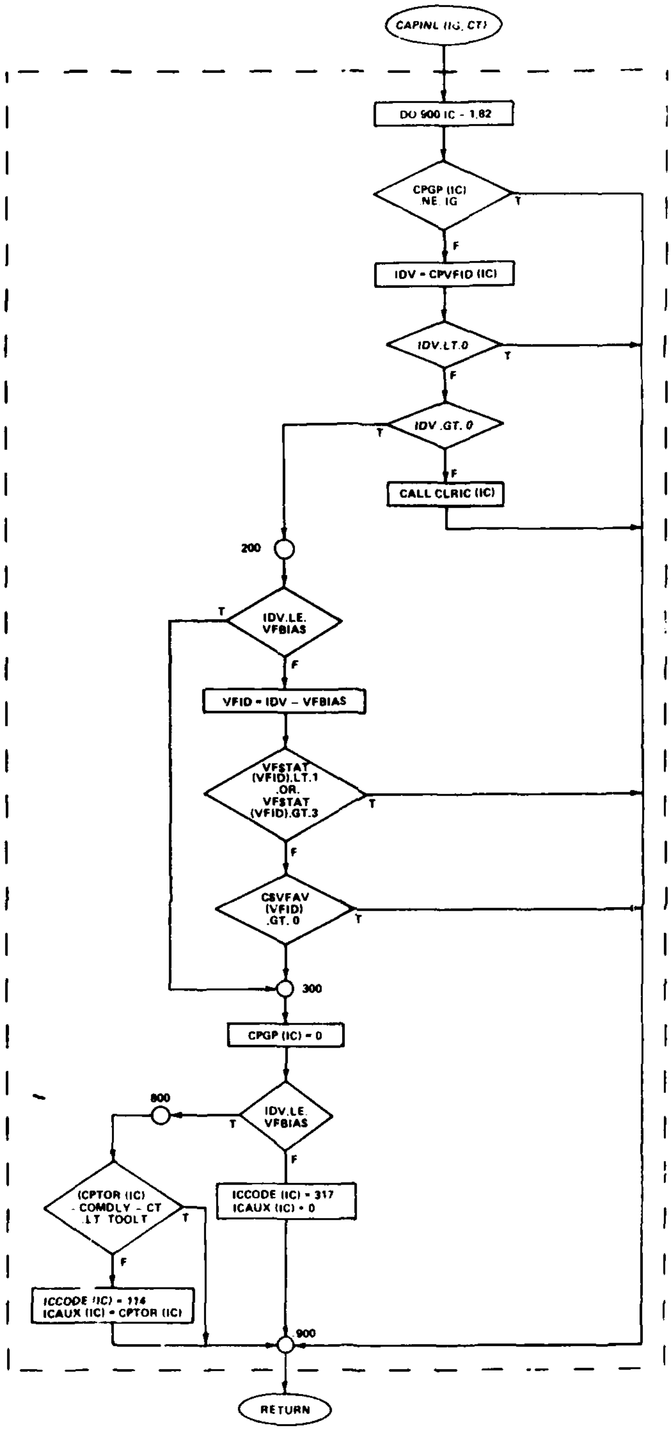


Figure 6-15. Subroutine CAPINL

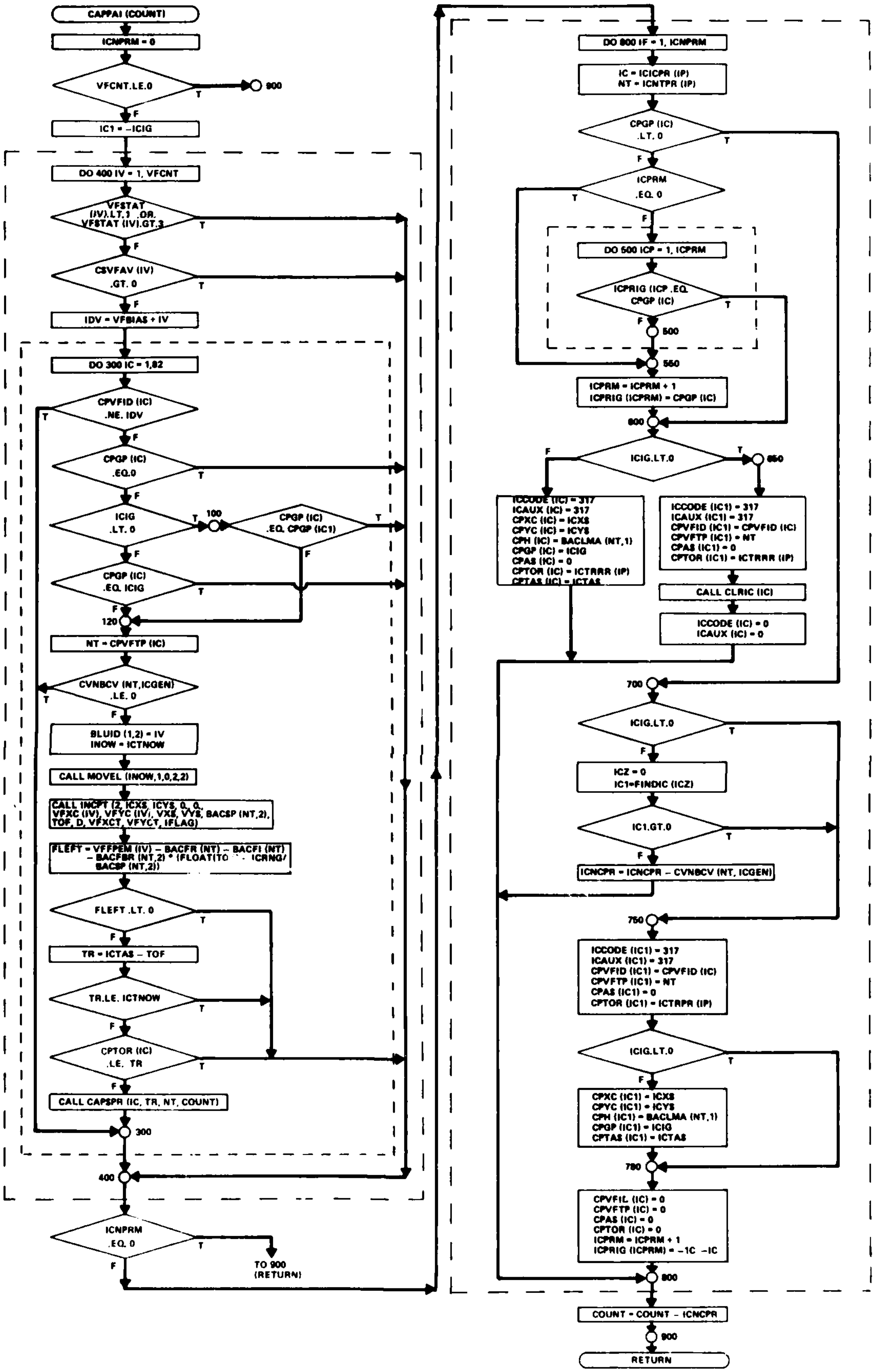


Figure 6-16. Subroutine CAPPAI

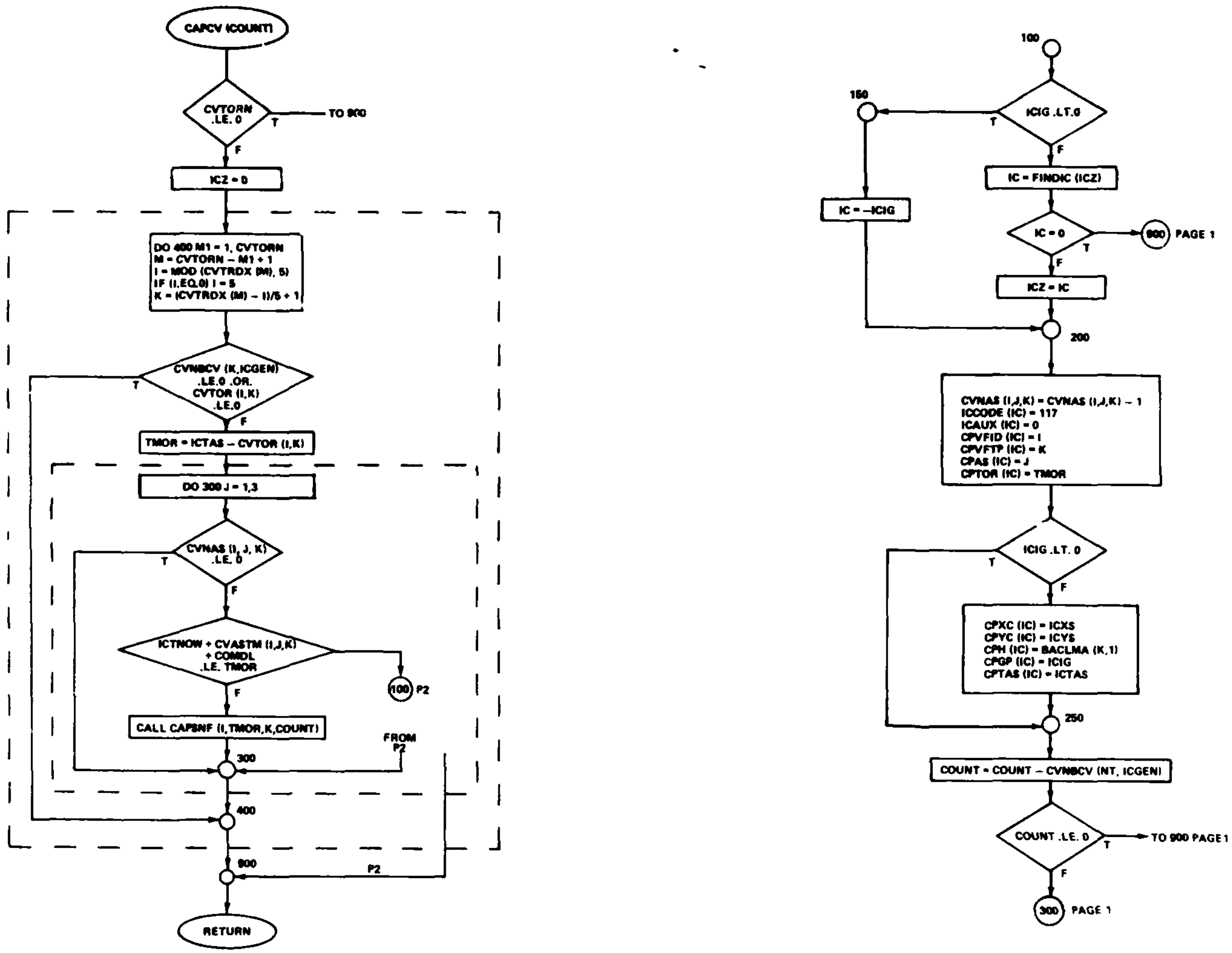


Figure 6-17. Subroutine CAPCV

as completely as possible. A flow chart of this subroutine is Figure 6-18. Subroutine CAPRIC is used to structure the messages, force orders, which are used to convey the plan to the battle force elements. A flow chart of this subroutine is Figure 6-19.

Subroutine CAPSNF maintains a list of unassigned fighters with time of response prior, but closest, to the current time. These fighters will be assigned in the event fighters with times of response in the future are not available. These fighters may not be able to cover the most stressing threat, but offer some capability if the threat does not maneuver optimally. A flow chart for this routine is Figure 6-20.

Subroutine CAPSPR maintains a list of assigned VF with latest, future, time of response for use in preemption planning. Subroutine CVLNCH calculates the time of response required to reach a specific position for each type of fighter on each carrier. Subroutine INTTBL calculates a table of worst cases to be used if limited knowledge is known about a group being reported. Subroutine SURMSG processes the surveillance messages and maintains information arrays about each group. A flow chart for this subroutine is Figure 6-21.

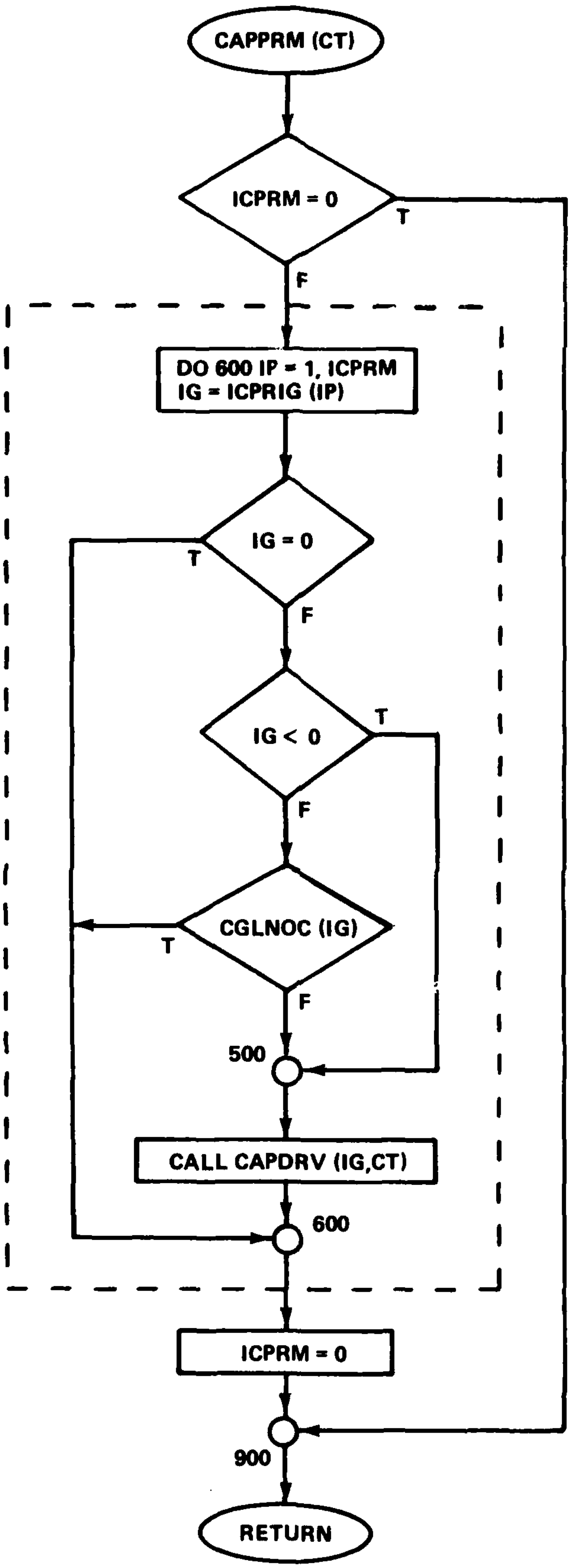


Figure 6-18. Subroutine CAPPRM

(rev. 10/15/82)

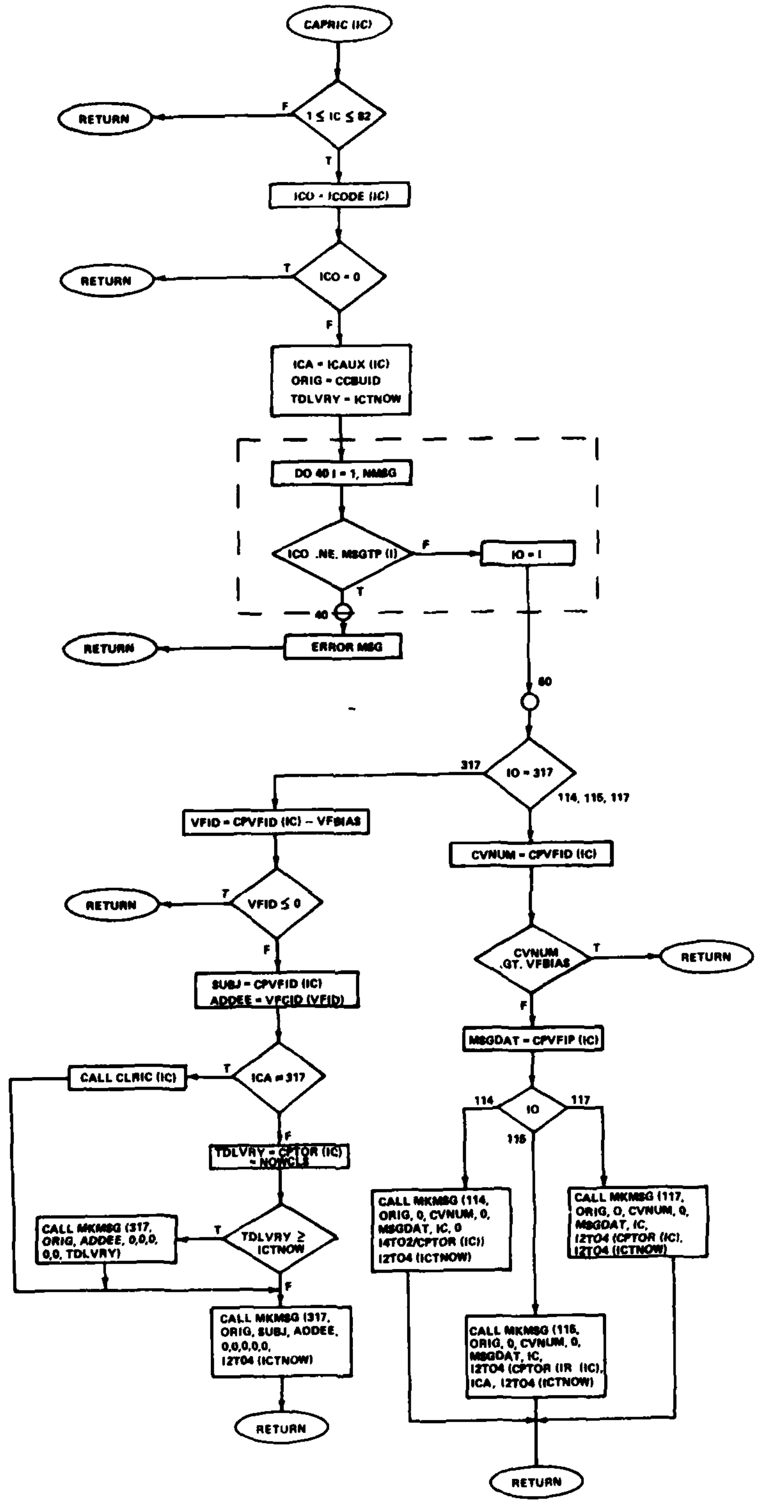


Figure 6-19. Subroutine CAPRIC

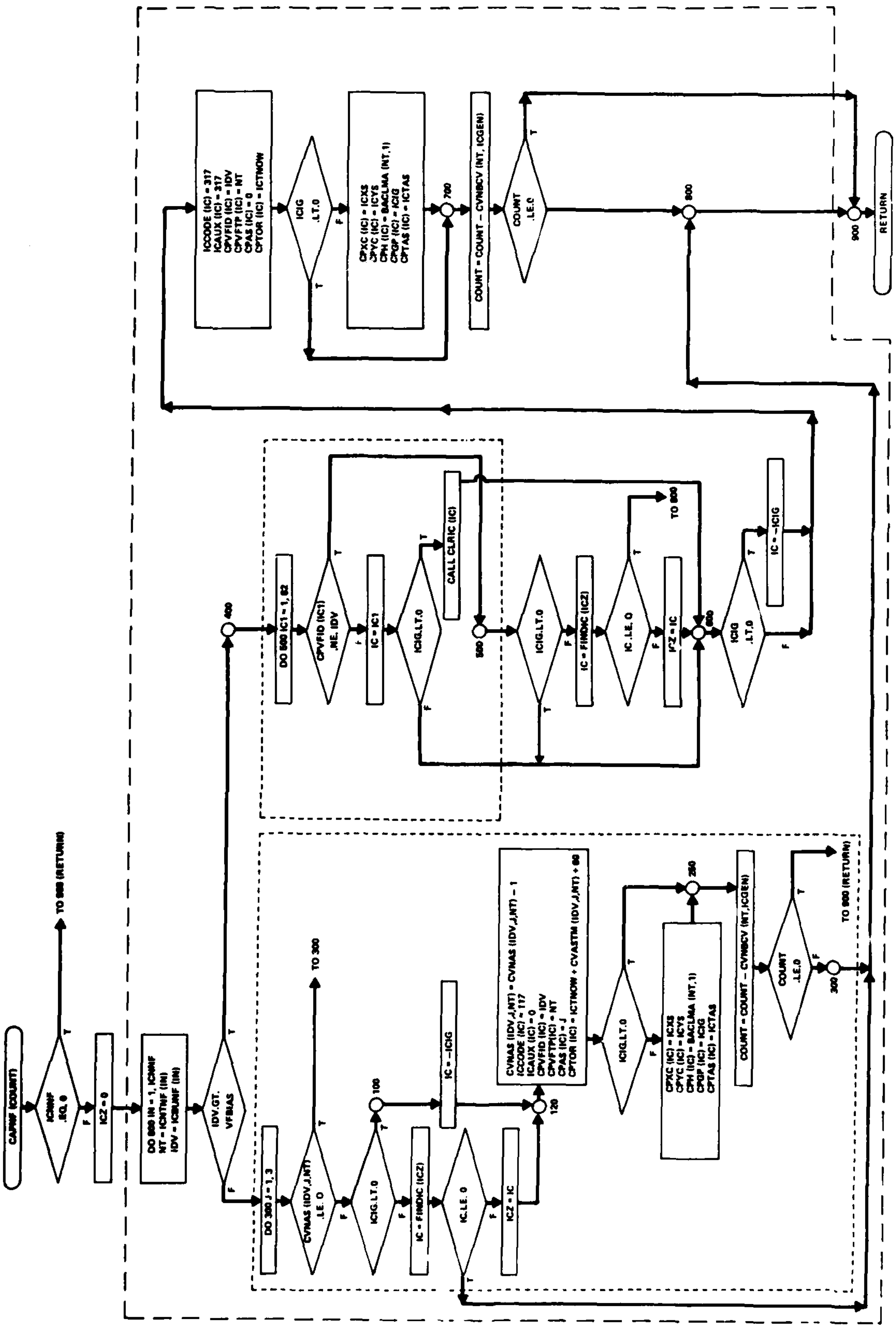


Figure 6-20. Subroutine CAPNF

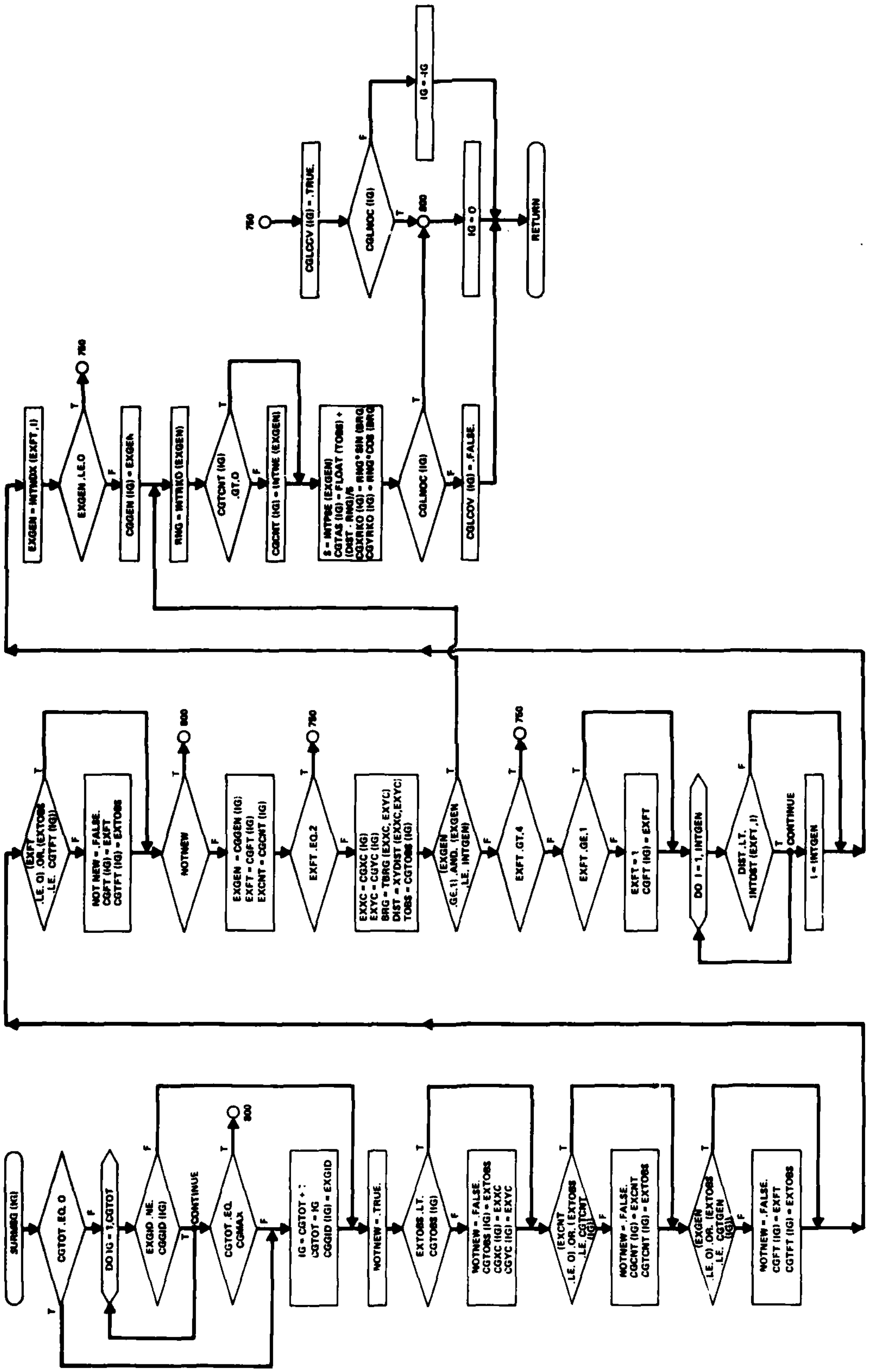


Figure 6-21. Subroutine SURMSG

7. AIR CONTROLLER MODULE

The Air Controller module simiulates the VAW or ship functions that vector interceptors. The function of assigning interceptors to targets is done in the Command Center module and is described in that section. In addition to vectoring VFs, the Controller acts upon requests by the VF for self intercepts and relays messages between the Command Center and the VFs.

Although the Controller maintains a list of detections (List A) via the Detect and Track module, the controller-related functions do not begin until the Command Center assigns VFs to it: Controller assignments are of two types - in combination with a target assignment (Message 108), or without a target assignment (Message 301). When only the assignment of VF to controller is being made, a check is made to see if the Controller can accept more assignments. List B is developed to keep track of the Controller's interceptors. If the Controller's capacity for interceptors (MAXVF) is exceeded, a "Controller Saturated" (Message 306) is sent to the Command Center. If the VF can be accepted, it is added to List B as Category 1 (unassigned to target).

If the Command Center wishes to assign to the Controller a VF with a target (Message 108), the Controller will have to begin vectoring the VF immediately. Therefore, the Controller's capacity to vector will be tested before accepting the assignment. If the Controller's capacity (MAXC) is exceeded, the "Controller Saturated" message is sent. If the VF is accepted, it is added to List B as Category 1. If the VF was previously assigned to the Controller and now a target is to be assigned (Message 101), a procedure similar to that with Message 108 is followed.

After a target assignment is made, the Controller will check the target position and compute the vector for the interceptor to fly, the controller has detected the target on its own radar, or the controller is aboard an NTDS Participating Unit (PU) and the target is being tracked by any other PU.

If intercept is feasible, the assignment is sent to the interceptor. If this is the initial message making the target assignment, the VF will be set to Category 2 (assigned to target) in List B. If the target makes a course change, a new vector must be computed. If the controller had contact on its own sensor, the Red course change comes from the Detect and Track Module. If the vectoring was being done from NTDS data, Message 010 triggers the new vector which is then sent to the interceptor.

The Interceptor module may respond by either accepting or rejecting the assignment. The Controller then sets the appropriate category and relays the message to the Command Center.

The remaining Controller functions operate independently of the foregoing functions. When the interceptor gains contact and takes self-control, the Controller puts it in Category 3 on List B.

If the VF detects a target while unassigned, it sends a "Self-Assign Request" (Message 103). The controller denies the request unless the target is classified "not engaged" on List A. It must also check the NTDS list (List C) if it is receiving NTDS data. If it is also classified "not engaged" in List C, the interceptor is given permission to attack. Also, if neither List A nor List C contain the target, permission is given to attack.

The remaining functions are receiving interceptor messages, making appropriate list modifications, and relaying the messages to the Command Center.

7.1 AIR CONTROLLER MODULE - GPSS LOGIC

The GPSS logic for the Air Controller Module is diagrammed in Figure 7-1. Two types of transactions enter the module:

- o Sensor copies are sent from the Detect and Track Module when a Red threat being tracked by an air controller (VAW or ship) changes course.

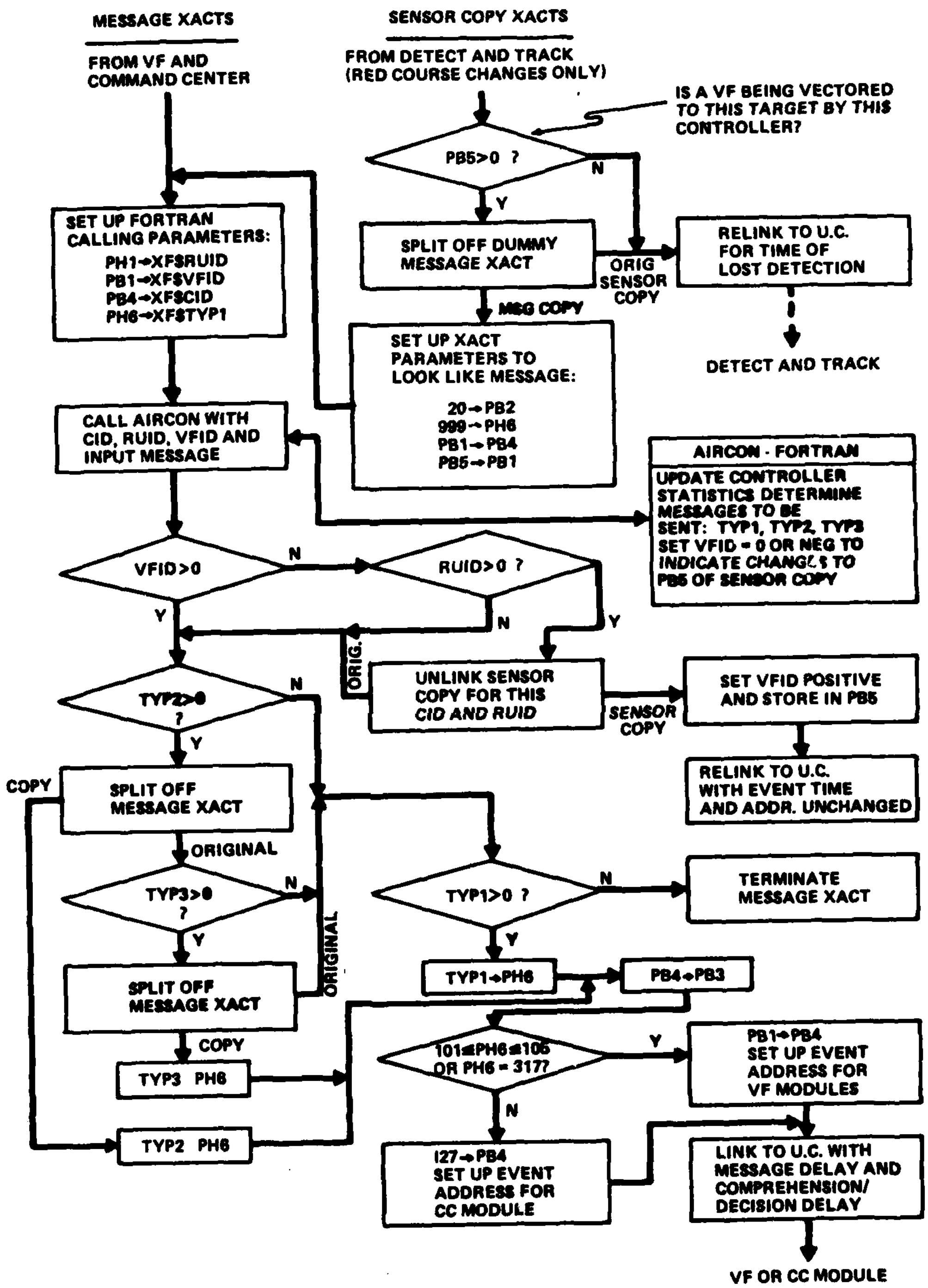


Figure 7-1. Air Controller Module (GPSS)

o Messages are sent from the VF Interceptor and Command Center Modules to report status or request action.

In order to process the two types of inputs in a similar manner, a dummy message transaction is split from an entering sensor copy and the sensor copy is linked back to the user chain. The dummy message transaction can then be processed just like the VF and Command Center messages. Note that no processing is required for a sensor copy if no VF is being vectored to it by the air controller who owns that sensor.

FORTRAN subroutine AIRCON is executed to determine the controller's response to the incoming message. The following arguments are used:

CID	Controller ID passed to AIRCON. This is the addressee (PB4) of all incoming messages and the PB1 parameter of the sensor copies.
RUID	Red unit ID passed to AIRCON. This is the PH1 parameter of sensor copies and of messages involving a Red unit. In a few instances, this savevalue may be set in AIRCON.
VFID	VF ID passed to AIRCON. This is the PB1 parameter of messages and the PB5 parameter of sensor copies. This savevalue is set zero or negative by AIRCON if the PB5 parameter of the sensor copy (for this CID and RUID) is to be reset or set, respectively.
MTYP	Message Type. This is used to pass the incoming message type to AIRCON and to return the first (lowest numbered) output message type.
MSG DATA	Message Data

After execution of AIRCON, VFID is tested for 0 or negative and RUID is tested for greater than 0, to determine whether the sensor copy for the given controller and Red unit is to be modified. If so, the sensor copy is unlinked; PB5 is modified; and the sensor copy is relinked to the user chain. Note that this sets PB5 to zero in situations where a VF was being vectored but no longer is; e.g. when the VF changes to a self-vectored intercept. Similarly if vectoring of a VF by the given controller is being initiated, PB5 is changed to the Blue ID of the VF being vectored.

Messages to be sent to the VF or Command Center Module are specified in MSG Matrix. The message transactions are then linked to the user chain with transmission and comprehension/decision delays.

7.2 SUBROUTINE AIRCON

Subroutine AIRCON analyzes the message to the controller and determines how to respond. The subroutine logic is diagrammed in Figure 7-2. The first page of the flowchart outlines the processing of messages sent by a VF. The second page shows the processing of messages from the command center and the Red course changes.

To support the decision-making, the module accesses a number of FORTRAN COMMON arrays. In addition, the module maintains the following arrays:

ACCTVF(N) Count of VF assigned to controller N.

ACCTVC(N) Count of VF being vectored by controller N.

These arrays are maintained as running totals that are incremented or decremented as VF are assigned/vectored or cease to be assigned/vectored. When new assignments are made by the Command Center, the pertinent count is compared with the corresponding maximum value from arrays ACMXVF(N) or ACMXVC(N) to determine whether the controller is saturated or if he can handle another assignment. Note also that the incrementing or decrementing of ACCTVC(N) occurs together with setting VFID negative or zero, respectively.

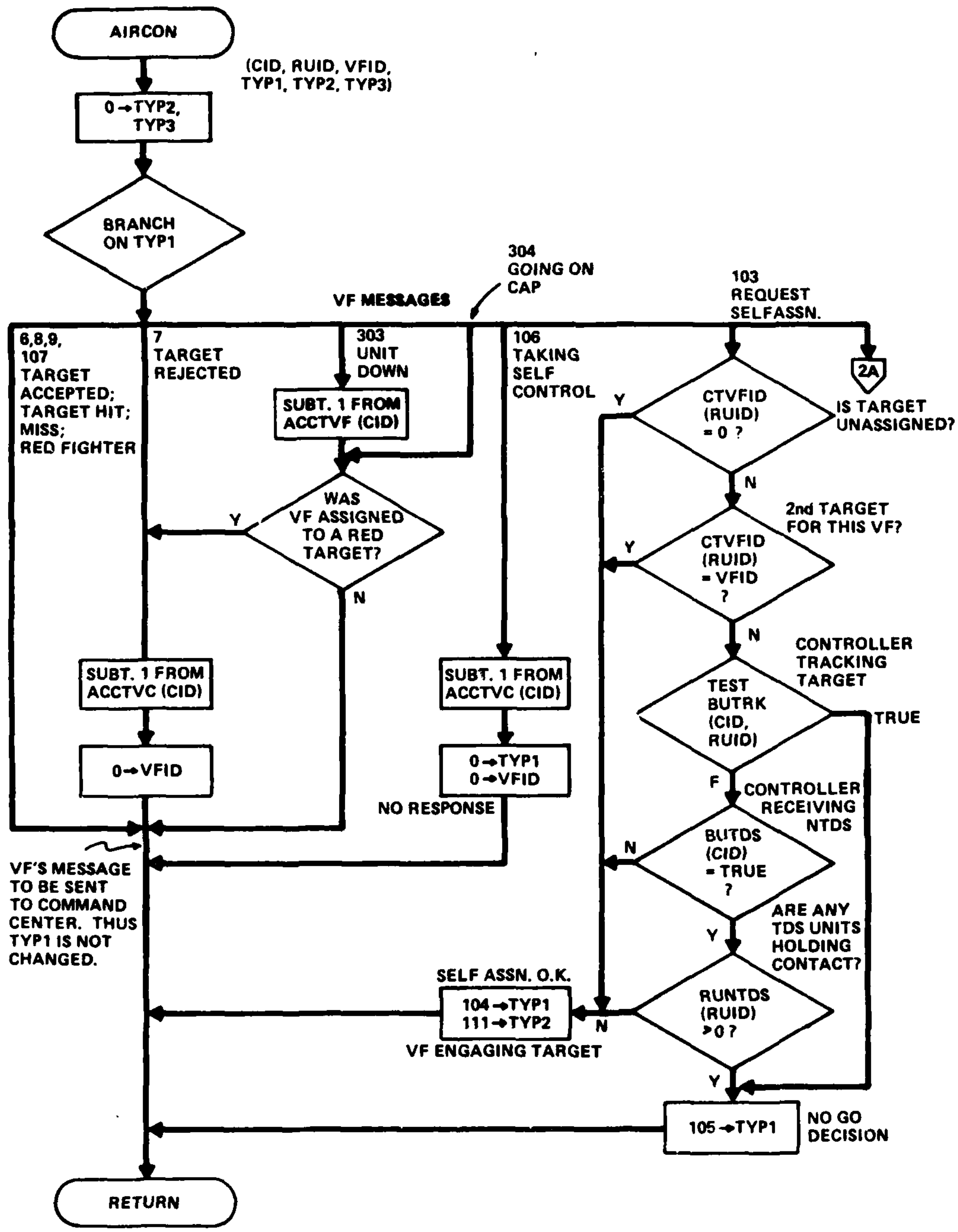
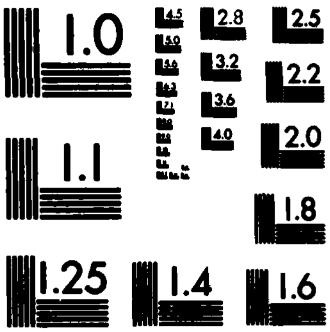
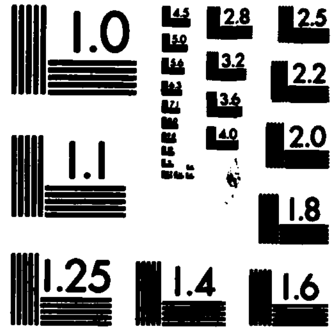


Figure 7-2. AIRCON Subroutine (Page 1 of 2)

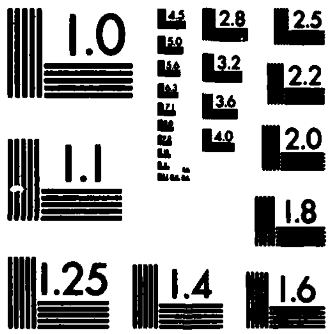
AD-A120 477 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) 3/4
(U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA WATERWHEEL
PROGRAM OFFICE R W COVEY ET AL. 15 SEP 82
UNCLASSIFIED N00014-81-C-0715 F/G 15/3 NL



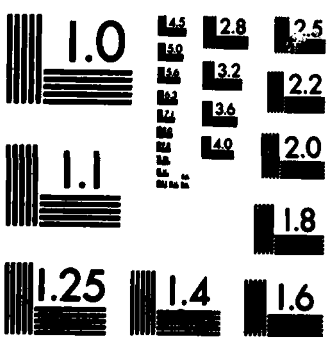
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



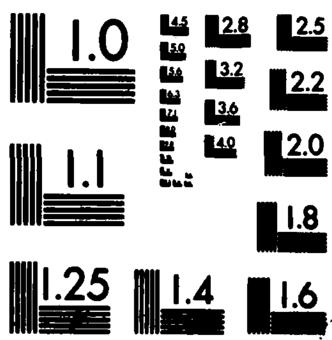
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

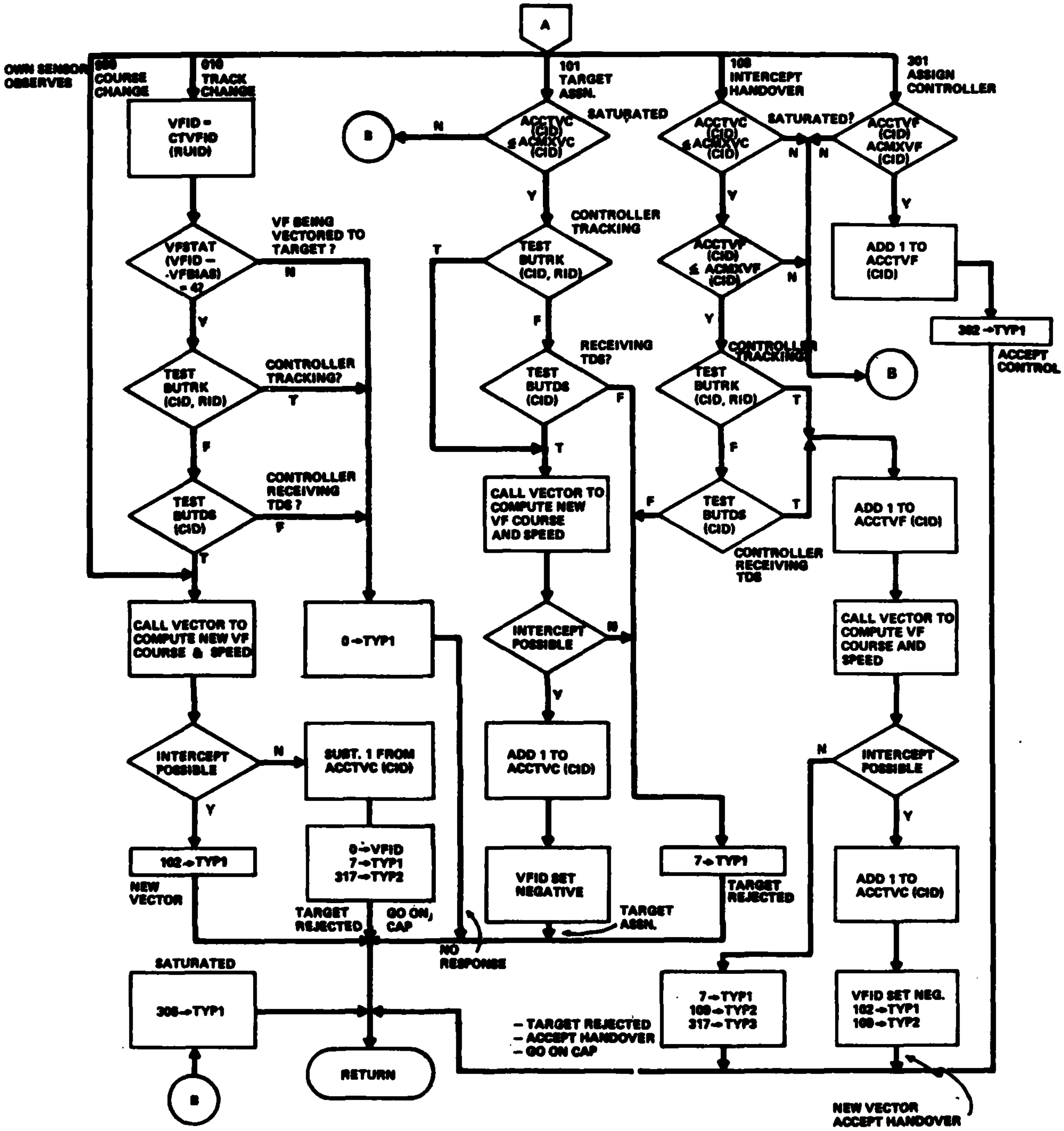


Figure 7-2. AIRCON Subroutine (Page 2 of 2)

8. INTERCEPTOR MODULE

8.1 GENERAL CONCEPT

The Interceptor Module performs all operations necessary to simulate the functions of the fighter-interceptors (VF). This module is complex because the interceptor moves, burns fuel, detects targets, can be controlled or operate on its own, and launches weapons.

Interceptors operate under the direction of a controller. By direction of the Command Center, the controller will normally assign an interceptor to a target and vector it. When the interceptor detects the designated target, it takes over control of its approach and attack. It selects the weapon and applies the launch strategy selected by the user. If the assigned Red aircraft turns out to be an escort fighter, the interceptor will have to engage it. If the interceptor loses the encounter, it is terminated. If the interceptor wins the encounter, it resumes CAP status at its present position. When the initial target is killed, the interceptor is made available for additional assignments if fuel and weapons permit. The interceptor may also initiate this sequence of events from its own detections, reporting its intent to the controller.

To facilitate handling the various conditions under which the interceptor may operate, a series of VF states (VFSTAT) have been defined. Table 8-1 shows the eight states. Two additional states, 0 and -1, are defined to indicate the return to the carrier and the VF being lost in the escort encounter. The interceptor can move at one of three speeds: (1) loiter, a speed that minimizes fuel consumption; (2) normal intercept, an intercept speed that does not burn fuel excessively; and (3) maximum, a high speed that will enable the fastest intercept (e.g., with afterburners). Because the simulation model must simplify many of the actions by the aircraft (such as turns and accelerations), the interceptor does not always move during the simulation. This is the situation during launch, CAP station, and escort encounter, VFSTATs 1, 3 and 8, respectively. Of course fuel is consumed even without movement as shown in Table 8-1. CAP status places the interceptor at a single point

Table 8-1. VF States

VFSTAT	1	2	3	4	5	6	7	8
Functions \ Description	CAP/DLI Launch	On Way To CAP Station	On Cap Station	Vectored Intercept*	Ready for Self Intercept	Self Intercept	In LAR	Dogfight with Fighter-Escort
Movement (Speed)	None	Loiter	None	Normal or Max	Normal or Max	Normal or Max	Normal	None
Fuel Consumption Rate	Warm-up & Take off (Total)	Loiter	Loiter	Normal or	Normal or	Normal or	Normal	Max
Radar	Off	On	On	On	On	On	On	On
Radar pattern	None	Sector	Omni	Sector	Sector	Sector	Sector	Sector
Control	CV	Controller	Controller	Controller	Self	Self	Self	Self
Time to next change	Input	Time to CAP (computed)	o Computed time to return to CV, or determined: o Self-detect o Assignments	Time to intercept point (computed)	Instantaneous	Time to LAR or Intercept point (computed)	Time to missile hit or miss	Input
Next State	2 or 4	3,4,or5	4,5,or 0	5	6	7 or 8	3 if hit 6 if miss	3 if win -1 if lose

VFSTAT = 0 Return to CV VFSTAT = -1 VF Hit *DLI or CAP representing an orbiting pattern. The radar search pattern is omnidirectional on CAP because the normal sector pattern would sweep a circle during its orbit. The actual encounter with the escort is not modelled, therefore the interceptor is stopped for the duration of the encounter, but burns fuel at the maximum rate.

The interceptor spends a finite time in most of these states. However, state 5 is used for computational purposes only and is occupied only momentarily.

The sequence of states will vary depending on what is happening in the game. Potential next states are listed in Table 8-1.

In the scenario the escort fighter is treated much like a bomber or stand-off jammer; it will fly a preplanned track that may look like a bomber track. The Blue sensors will not distinguish it from the other aircraft types, and attacks will be initiated as if it were a bomber. The classification is corrected when a "fighter" message is received from an engaged VF. If a dogfight takes place, an additional delay occurs, and an exchange ratio is invoked to determine who wins.

Interceptor Events

Events generally happen in a predictable order, but there are alternate paths, and some events may pre-empt others.

After the controller receives the command center assignment for an interceptor, it sends the command to start the intercept (VFSTAT=4) and includes the course and speed. After a suitable communication delay, the interceptor replys with WILCO. The next event should be detection of the assigned target by the interceptor. In case the interceptor does not detect the assigned target, it will return to CAP status (VFSTAT=3) when it reaches the designated intercept point.

The path of events starting with detection by the interceptor will now be traced. The first event in this path is VF decision. Basically the VF Decision event requires a Help Block to:

- pick best target (unless already assigned)
- compute intercept vector
- compute time to launch the intercept missile
- compute the results of an escort encounter, if any.

These functions will be further explained in succeeding paragraphs.

Depending on which type of event occurs, escort encounter or intercept missile launch against a bomber, that path will be followed. Assuming an escort encounter occurs, a message is sent to the controller and Command Center that the interceptor is engaging an escort. The interceptor stops its motion for the duration of the encounter, which is input based on some average time derived from other studies. At the end of this time the winner is chosen, again based on probability inputs derived from other analyses. If the interceptor loses it is terminated and a "VF Lost" message is sent to the controller and Command Center. The time to recognize this loss is controlled by a recognition delay input by the user. If the interceptor wins, it returns to CAP status at that position, ready for another assignment unless its fuel and ammunition status forces its return to the CV.

If the first event to occur after the VF Decision event is the launch of the intercept missile (Launch AIM), then the flight time of the missile must be computed to determine the following event, which is a hit or miss by the missile. Hit or miss is determined from input probabilities of hit for the missiles of interest. A miss results in a message "Target Missed" and a return to the VF Decision event to determine if additional attacks can be made. If the missile hits the target, the message "Target Hit" is sent and the interceptor goes onto CAP station at its present position. At that time the interceptor is available for reassignment by the controller or may take up additional intercepts on its own.

The Command Center initiates commands to launch DLI or CAP. VFs are brought into the game and initialized (given an ID, a controller, coordinates, etc.). DLIs receive their vectoring instructions from the Controller right from the start. CAP bound aircraft will receive their initial vector to CAP station in the Interceptor Module. While flying to its CAP station (VFSTAT=2) the Command Center and Controller can use the VF as an available interceptor if desired. Also upon starting to DLI assignment or toward CAP station, Red copies must be created for the sensor.

Upon arrival at CAP station, detection must be recomputed because of the detection pattern change. Also the time to be spent on CAP station is computed. If this time expires without the Interceptor being used, it returns to the carrier and all its sensor copies are terminated.

8.2 INTERCEPTOR MODULE GPSS LOGIC

Three types of transactions are processed by the VF Interceptor Module.

- Messages are received from and sent to the Air Controller Module. These are addressed to a specific unit and from a specific unit.
- Sensor transactions are sent from the Detect and Track Module when a VF starts tracking a new target, or when a target being tracked changes course. The VF module sends sensor transactions to Detect and Track for recomputation of detection times when a VF goes on CAP, leaves CAP, or changes course. Sensor transactions are used within the VF Module to schedule events related to a specified target that is being tracked.
- VF control transactions are used to schedule events that are unrelated to any particular target. This includes:

-- Arrival on CAP station.

This event is scheduled by the CV Module when a CAP is launched. The event is cancelled if the VF is diverted for an intercept before reaching the CAP station.

-- Arrival at collision intercept.

This event is scheduled in the VF Module when a VF is vectored to an intercept by an air controller. This event should never occur, since it is always cancelled when the VF detects the target on his own sensor. The event is scheduled as a safeguard for a situation where the target changes course, but the new vector is not received by the VF because of jammed communications. If the VF gets to the intended intercept point without detecting the target, he goes on CAP until he's assigned a new target or he detects and self-assigns a target.

-- Return to CV

This event is scheduled in the Blue Scenario or VF Modules when a VF is on CAP at the beginning of the game or goes on CAP during the game, respectively. The return time is based on the fuel the VF has when he goes on CAP and the projected consumption rate. This event is cancelled if the VF goes on an intercept, since fuel levels are watched as part of each intercept event.

FORTRAN subroutine, VFCALC, performs the computations for the VF Module. It is executed when a message is received or an event occurs. VFCALC is a FORTRAN subroutine called by GPSS HELPC HLPRTN. The following arguments are used:

VFID VF ID passed to VFCALC. This is the addressee (PB4) of all incoming messages and the PB1 parameter of sensor and VF control transactions.

RUID Red unit ID passed to VFCLC. This is the PH1 parameter of all sensor transactions and of messages involving a Red unit.

CEVT Event type. This is used to pass the event or message type to VFCALC for the event that is currently occurring or the message that is being received. VFCALC also uses this savevalue to return the event type of the next event that VFCALC has scheduled.

TIME This is used to pass the current clock time to VFCALC. VFCALC returns the time of the event (EVT) to be scheduled via the same savevalues.

CAPID Identification of the index to CAP station data.

NETYP Message Event Type.

Figure 8-1 diagrams the GPSS logic for the VF Module. The processing of the three transaction types has been separated, and is shown on individual pages. VFCALC is called separately for the three transaction types.

Incoming messages from the Air Controller Module enter the VF Module on Page 1 of the flowchart. VFCALC is called to update FORTRAN data arrays and determine the messages to be sent and the next event to be scheduled. The program then branches based on the next event that is to occur.

Depending on the next event, sensor and/or VF control transactions may be unlinked and used to schedule the next event or terminated. Messages are formatted and linked to the user chain starting at location A. This same segment of code will be used for all messages being sent by a VF.

Figure 8-1, Page 2 diagrams the processing of sensor transactions. Here, VFCALC may be called for transactions sent from the Detect and Track Module or for transactions representing events scheduled by the VF Module. The test for current event 902 (No Response to Self-Assigned Intercept) has to be performed before calling VFCALC, because the messages 103, 104 and 105 are unlinked only when and if event 902 occurs. The events on which the branching takes place after caling VFCALC are future events that have not yet occurred.

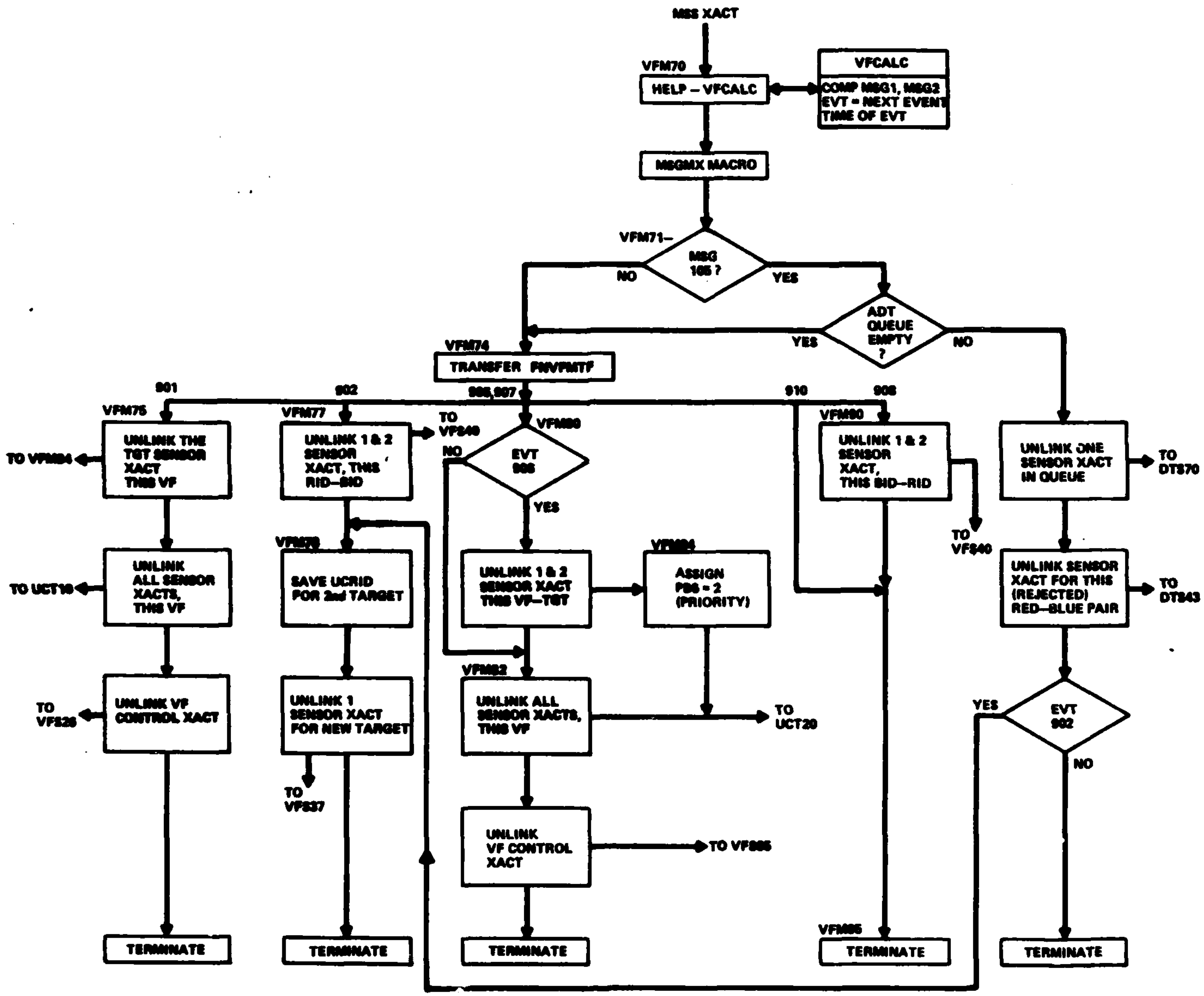


Figure 8-1. VF Module GPSS (Message Transactions) (Page 1 of 3)

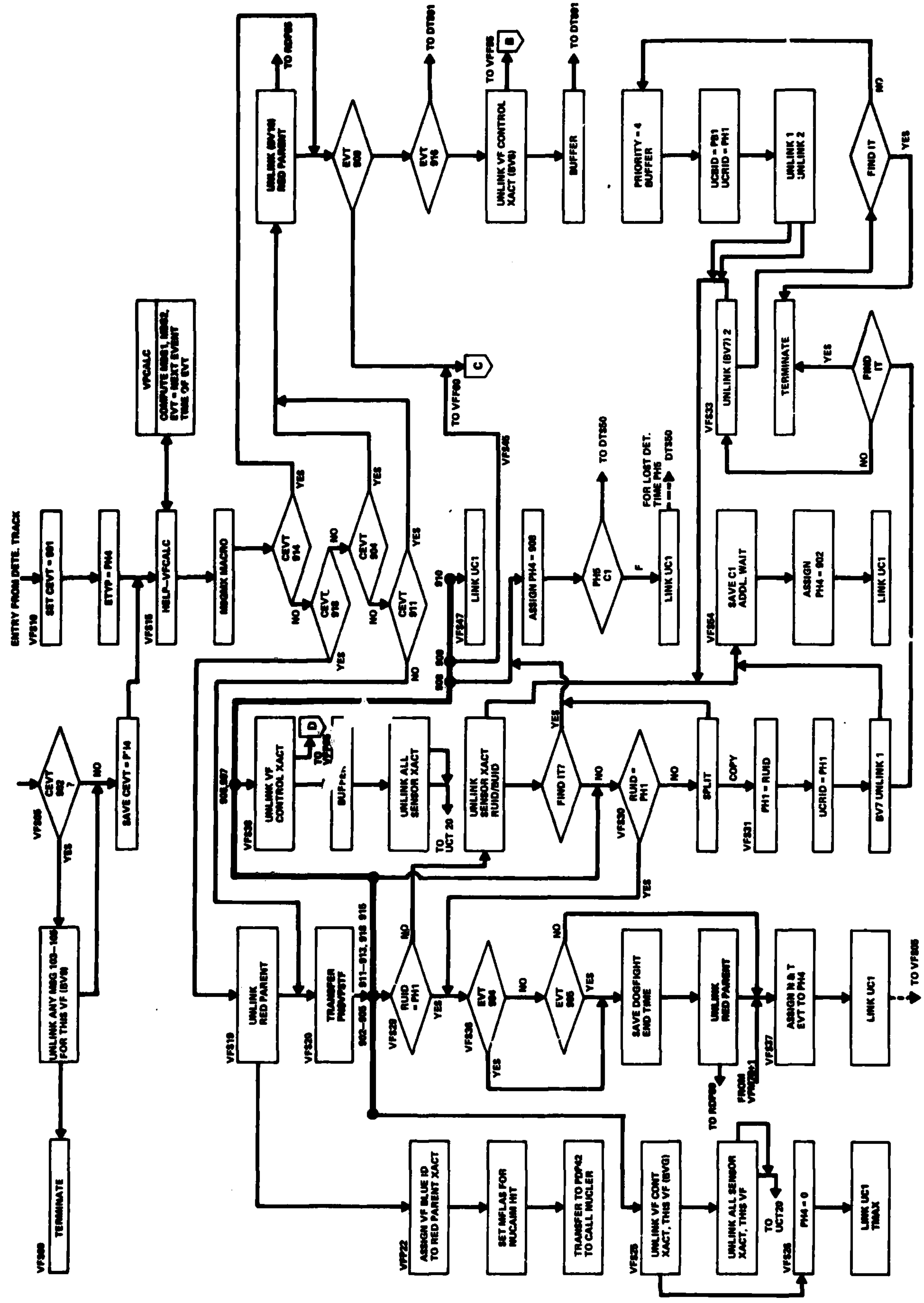


Figure 8-1. VF Module GPSS (Sensor Transactions) (Page 2 of 3)

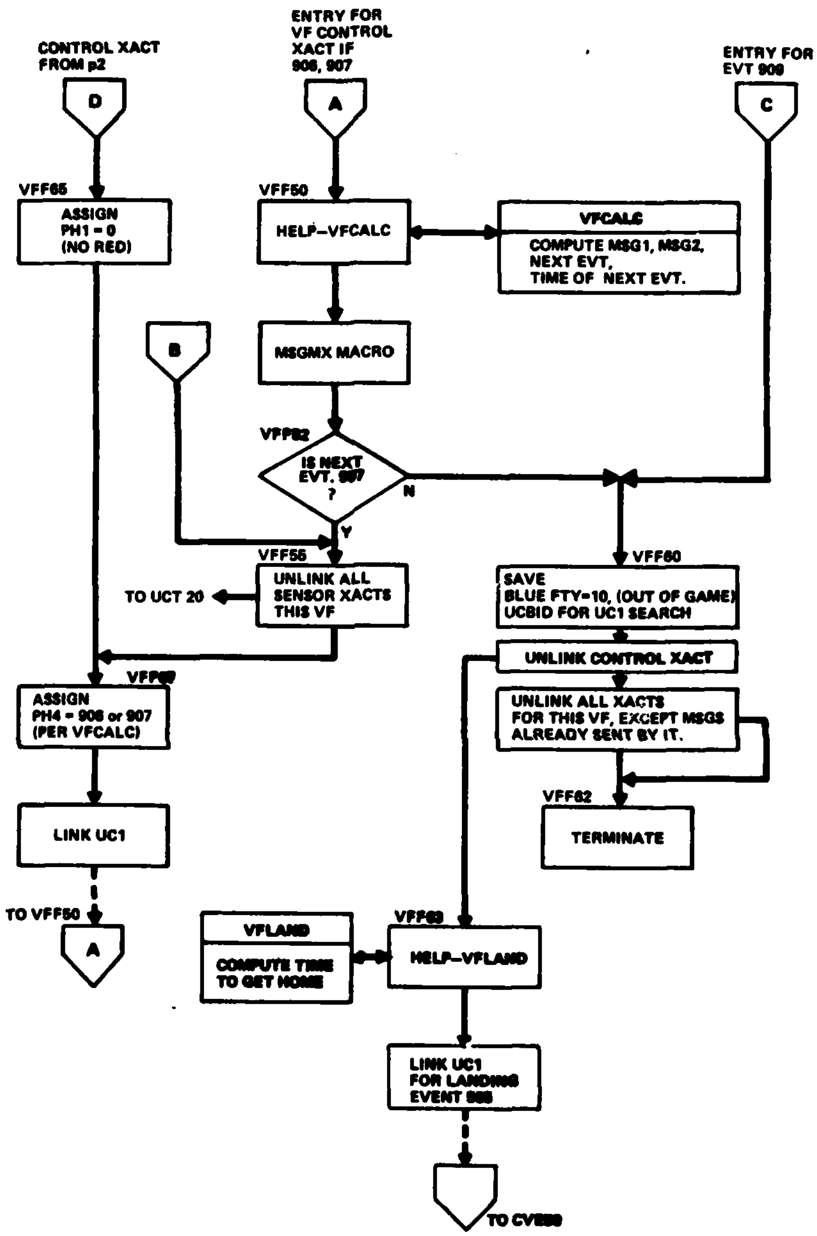


Figure 8-1. VF Module (VF Control Transactions) (Page 3 of 3)

Figure 8-1, page3, diagrams the processing of VF control transactions. Event types 906 and 907 are scheduled via VF control transactions and addressed to this location for processing when the event occurs.

It should be noted that some future/next events may be scheduled by more than one current event. Event 907 (Return to CV), for example, is shown as a next event on each page of the flowchart. In each instance the event is scheduled via a VF control transaction and addressed for processing (when and if the event occurs) to location D on the flowchart Page 3. The setting up of the control transaction and the other related processing differs somewhat depending on the type of transaction (message, sensor or VF control) with which the current event was scheduled.

8.3 FORTRAN SUBROUTINES

Subroutine VFCALC

Subroutine VFCALC analyzes the current (incoming) event, updates the COMMON data arrays, and determines the messages to be sent, the next event that will occur, and the time of the event. A flowchart of the subroutine appears in Figure 8-2.

Subroutine VECTOR (Figure 8-3)

A generalized vectoring subroutine enables a VF to intercept a target. It may be used for both decision making and controlling. The Command Center, Controller, or Interceptor itself may use it. Subroutine VECTOR is given a target and an interceptor and it determines if an intercept can be made and when it will occur. VECTOR selects the speed that the interceptor will fly and determines if it has adequate fuel. VECTOR calls subroutine INCPT, which computes intercept time, position and interceptor course.

The first operation is to determine if an intercept can be made. Subroutine INCPT is called and SPEED = 2 (normal intercept speed) is used first. If no intercept is possible at the normal speed, the higher intercept speed is attempted. If an intercept is possible, the fuel is checked to see if there will be enough. Because this subroutine makes tentative decisions while testing, status values may be changed only after the decision is made. Thus FREMC is a temporary value for testing and FREM is actual fuel remaining.

If a tanker or tankers are available it may be possible to extend the VF's time in the air. It will be permitted to burn its reserve fuel (FRES) under the assumption that a tanker will refuel it before the reserve would be needed. Use of the reserve is permitted one time only, and only until the tanker supply is exhausted. Tanker movement is not modelled.

If an intercept is possible, the VF may return to CAP status if it has enough fuel for a shorter intercept, or it may return to the carrier immediately if fuel is too low. When no intercept is possible the target remains on the unassigned list.

Subroutine INCPT

INCPT is a two-dimensional intercept routine. It computes time and distance to intercept, intercept coordinates, and intercept speed components. It has two modes of entry with respect to the interceptor speed. Mode 1 uses the target speed resolved into its X and Y components as it would come from the status arrays. In Mode 2 the speed is a single component as it would come from the characteristics arrays. Routines other than VECTOR may use INCPT and use different outputs than VECTOR requires.

Subroutine MOVEL

MOVEL will update to the current time the positions of all Red units and all VFs specified in the input to the subroutine. The VF's fuel status will be similarly updated at the same time.

The VFs are moved first. Those VFs that have been in a non-moving state need not be moved, but their fuel status is updated. It is also possible that MOVE may be entered with no time having elapsed since the last update, in which case no updates are performed.

After all designated VFs are updated, the Red units are updated to current time. The procedure is similar to that for the VFs.

Subroutine WEAPON

Figure 8-5 shows the tests that are applied to select among the four weapons. The Type 1 or 2 weapon is the most capable; 1 is nuclear, 2 is conventional; Type 2 can be salvo launched. Type 3 is second most capable and can be salvo launched. Type 4 has the least capability, is used only at close range, and must be launched singly. Figure 8-6 defines the geometric quantities used in the tests. Salvo launch policy for Types 2 and 3 are provided by the user input.

When salvo launches are specified, they are used in preference to single launches provided there are at least two missiles available. A salvo is always two. Type 1 or 2 missiles are selected first unless the present range to the target (RVFTT) is within LAR of a Type 3 or less than the minimum launch range of the longer-range type.

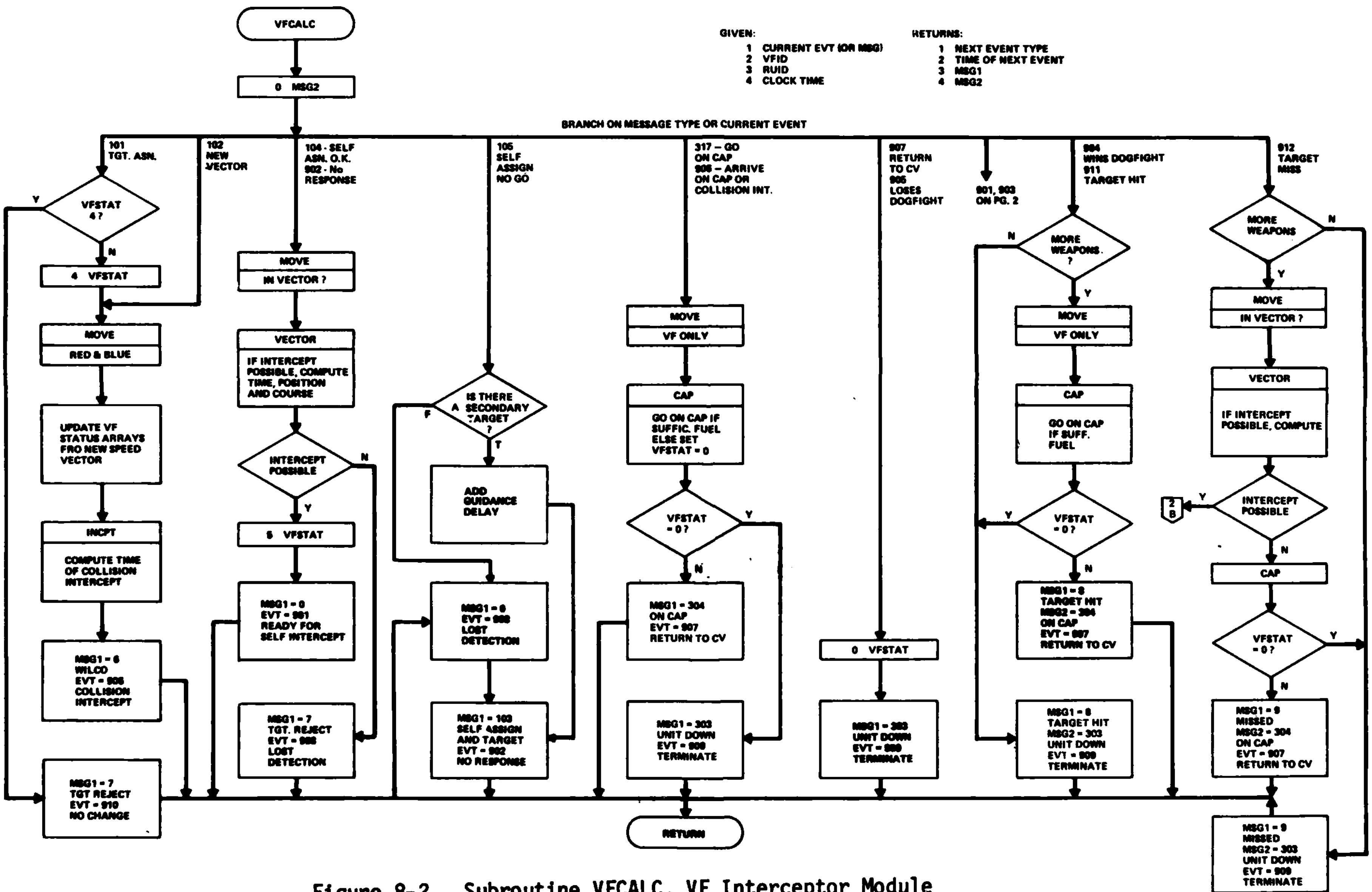


Figure 8-2. Subroutine VFCALC, VF Interceptor Module
Page 1 of 2)

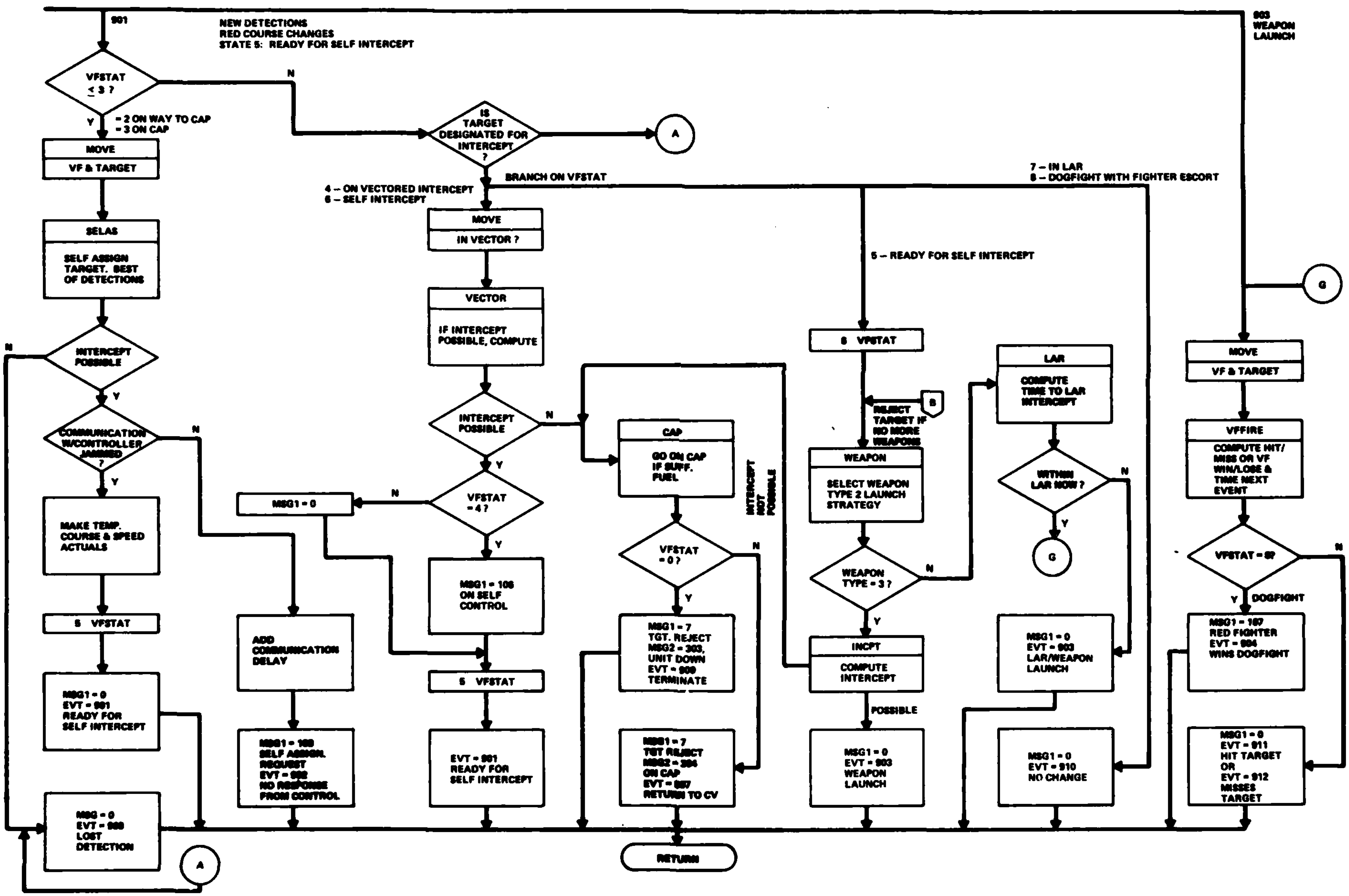


Figure 8-2. Subroutine VFCALC, VF Interceptor Module (Page 2 of 2)

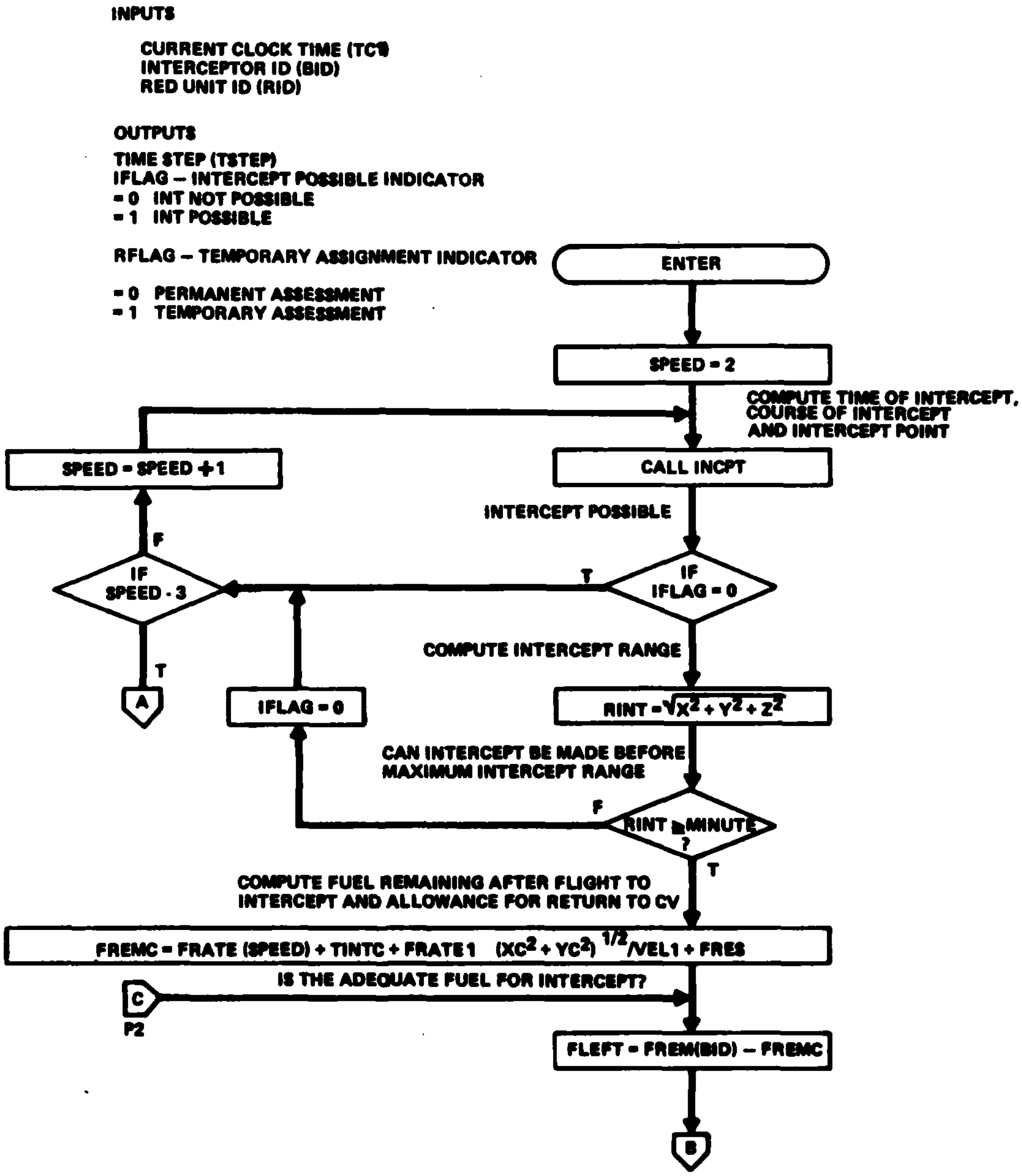


Figure 8-3. Subroutine VECTOR (Page 1 of 2)

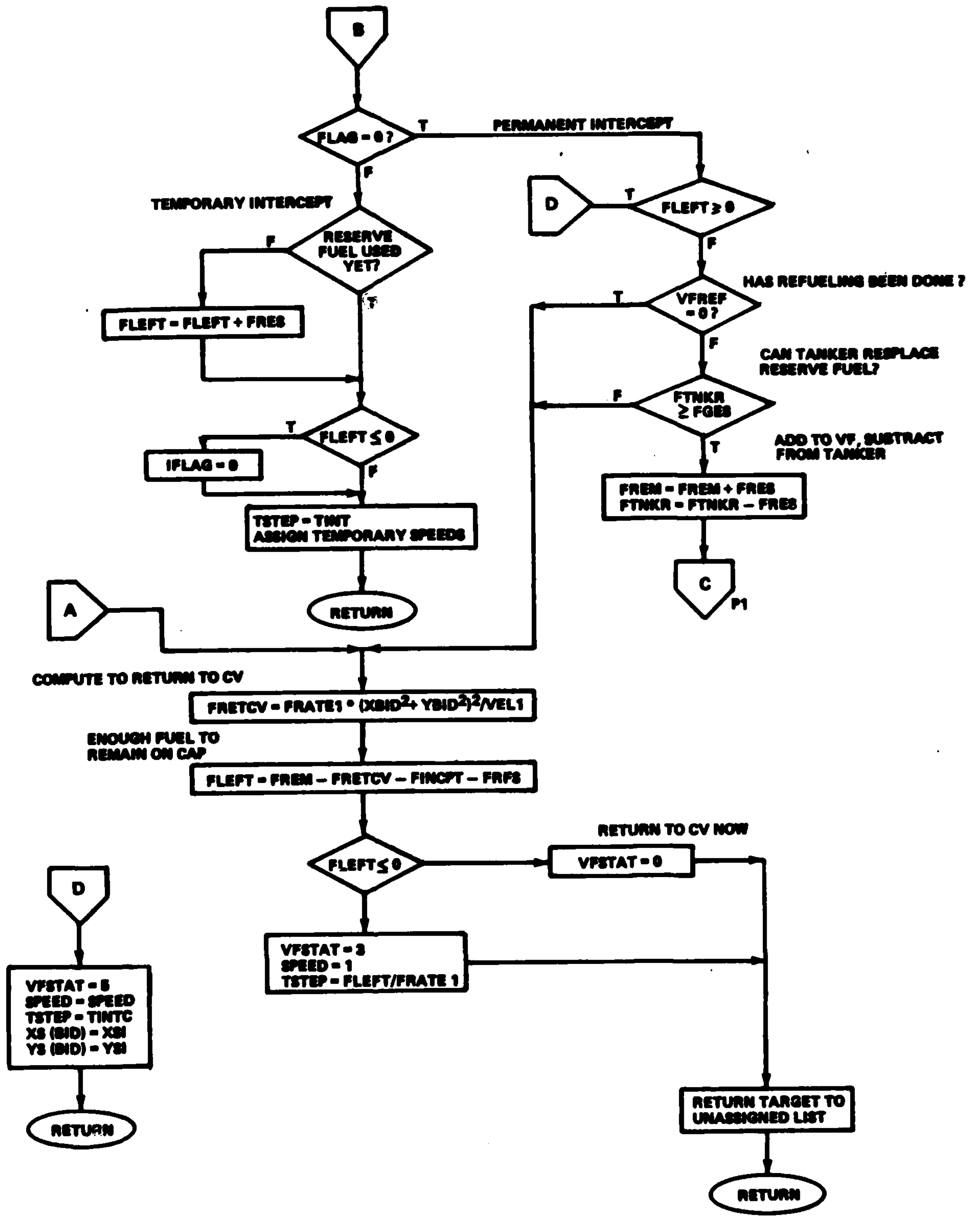


Figure 8-3. Subroutine VECTOR (page 2 of 2)

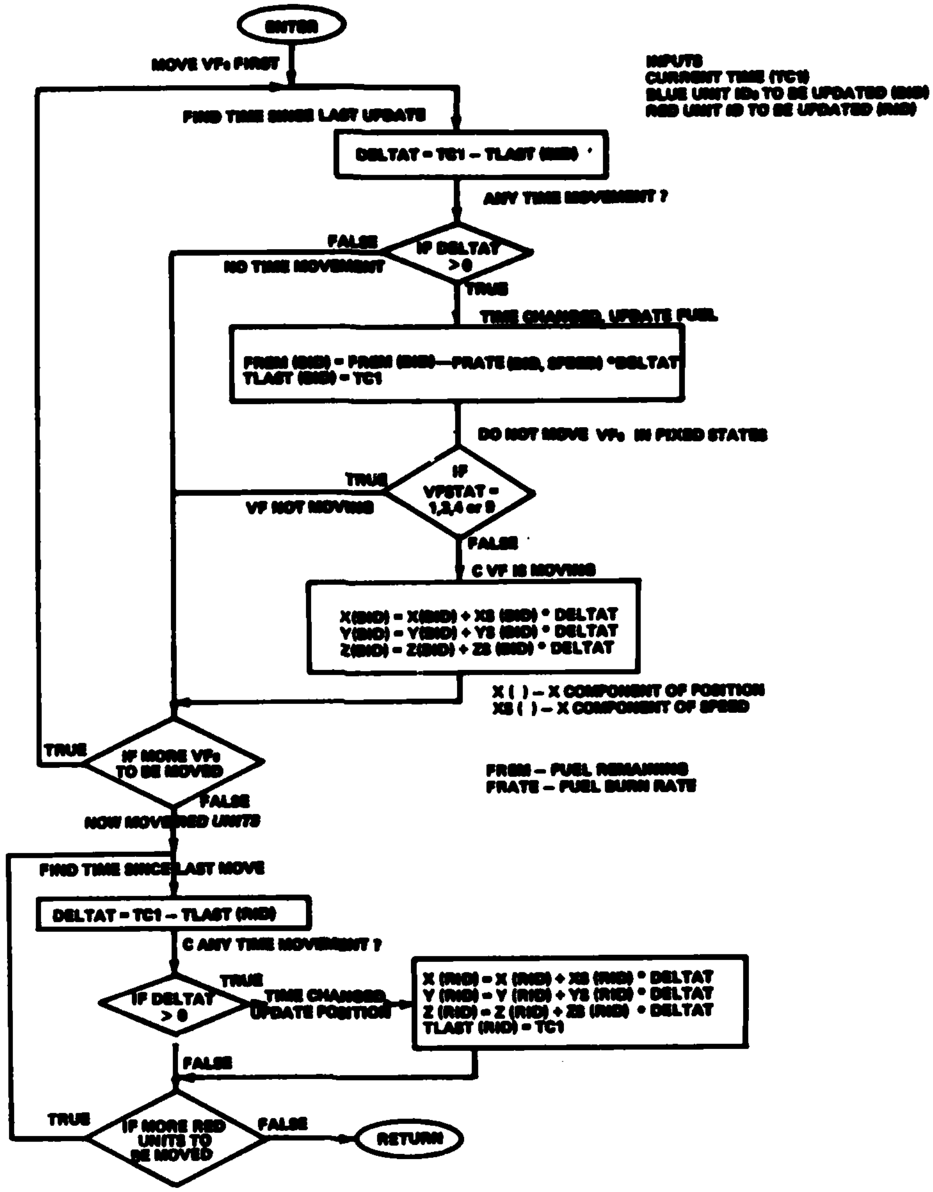


Figure 8-4. Subroutine MOVEL Updates Blue or Red Positions and VF Fuel to Current Time

HAV1 = True, if count of Type 1 Missiles 0 (Type 1 is Nuc)
HAV2 = True, if count of Type 2 missiles 0
HAV3 = True, if count of Type 3 missiles 0
HAV4 = True, if count of Type 4 missiles 0
SALV2 = True, If (count of Type 2 missiles 1). and.
(Doctrine for Type 2 is SALVO)
SALV3 = True, If (count of Type 3 missiles 1). and.
(Doctrine for Type 3 is SALVO)
NUCFLG = True, if set by prior routine that made the target assignment.
Indicates that a Nuc is the preferred weapon, if available.

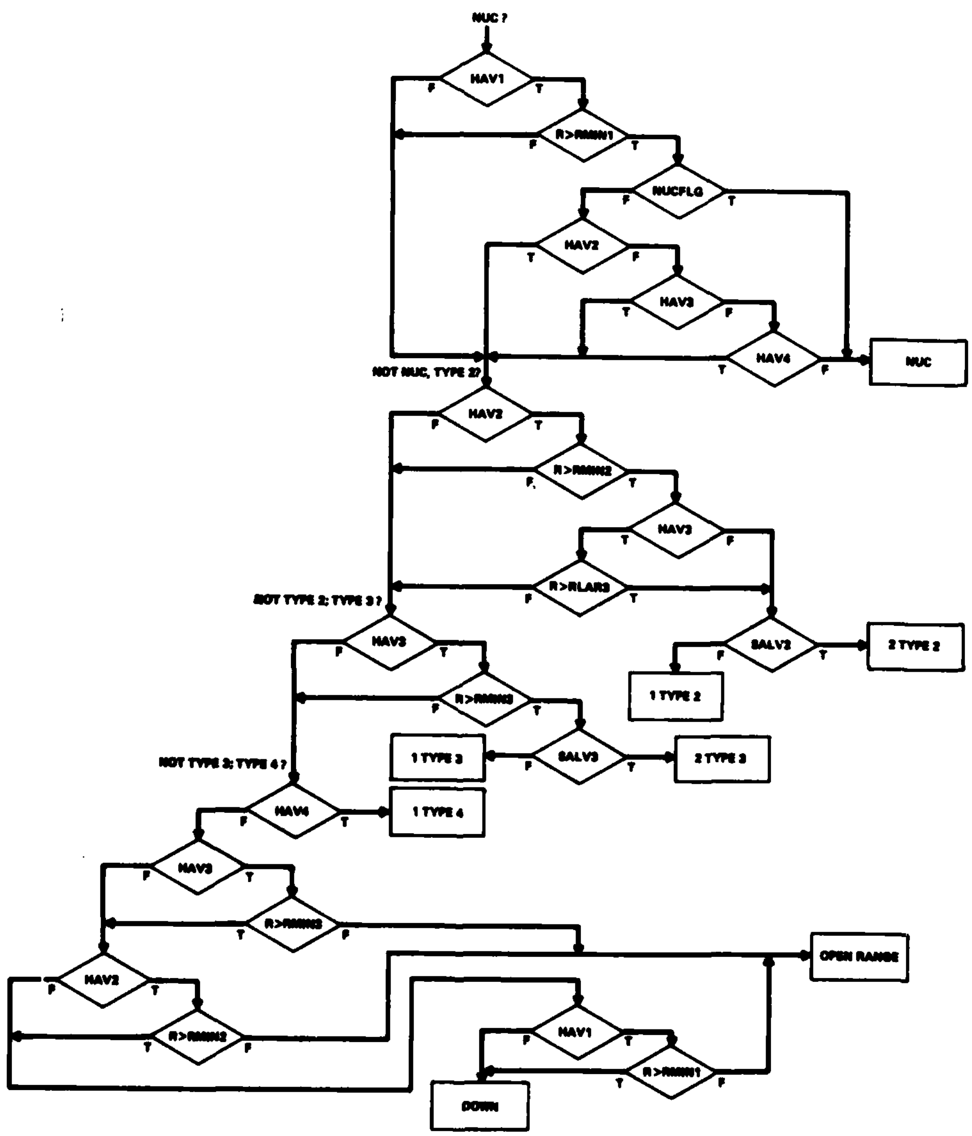


Figure 8-5. Weapon Selection Logic

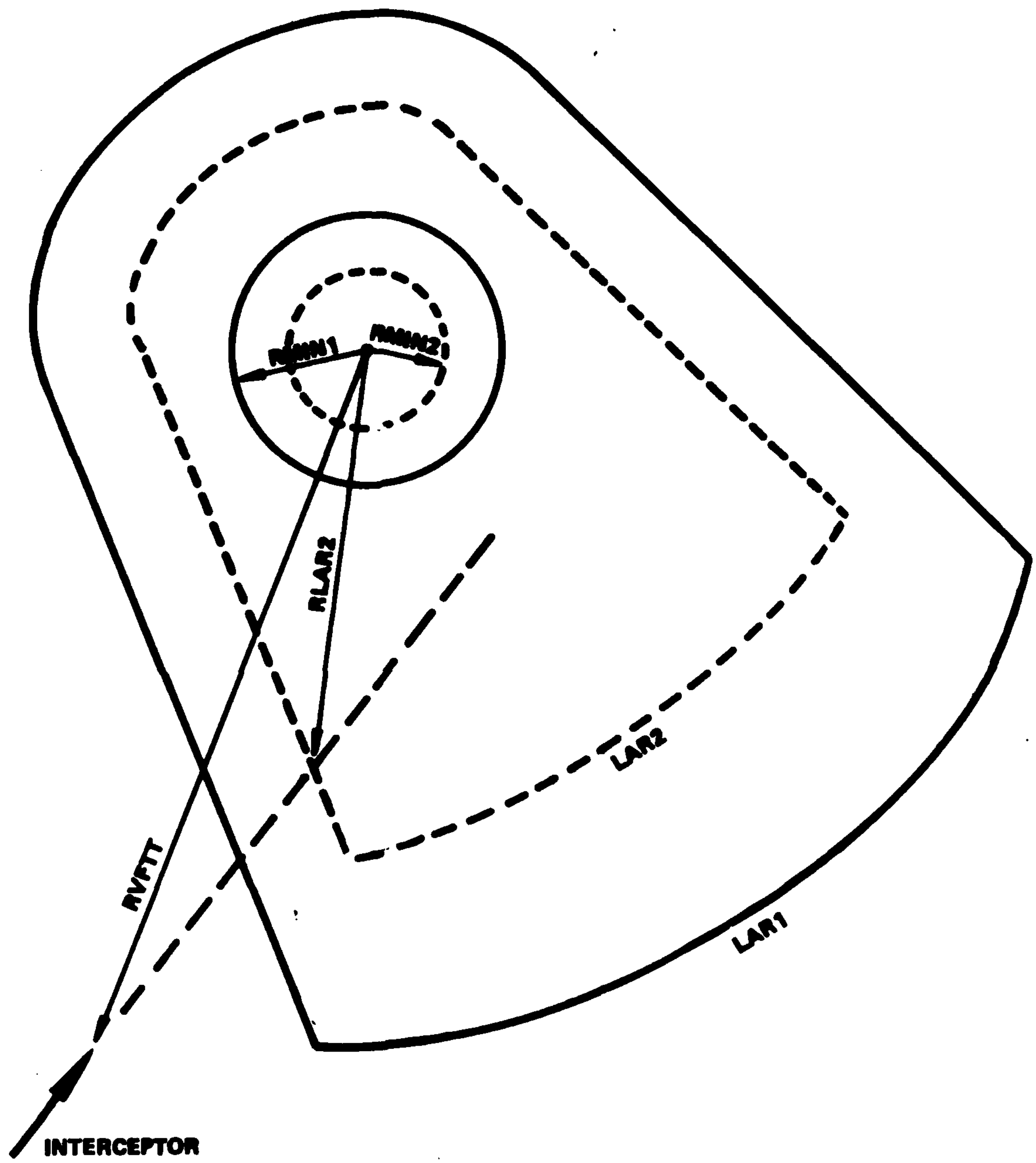


Figure 8-6. Weapon Selection Geometry

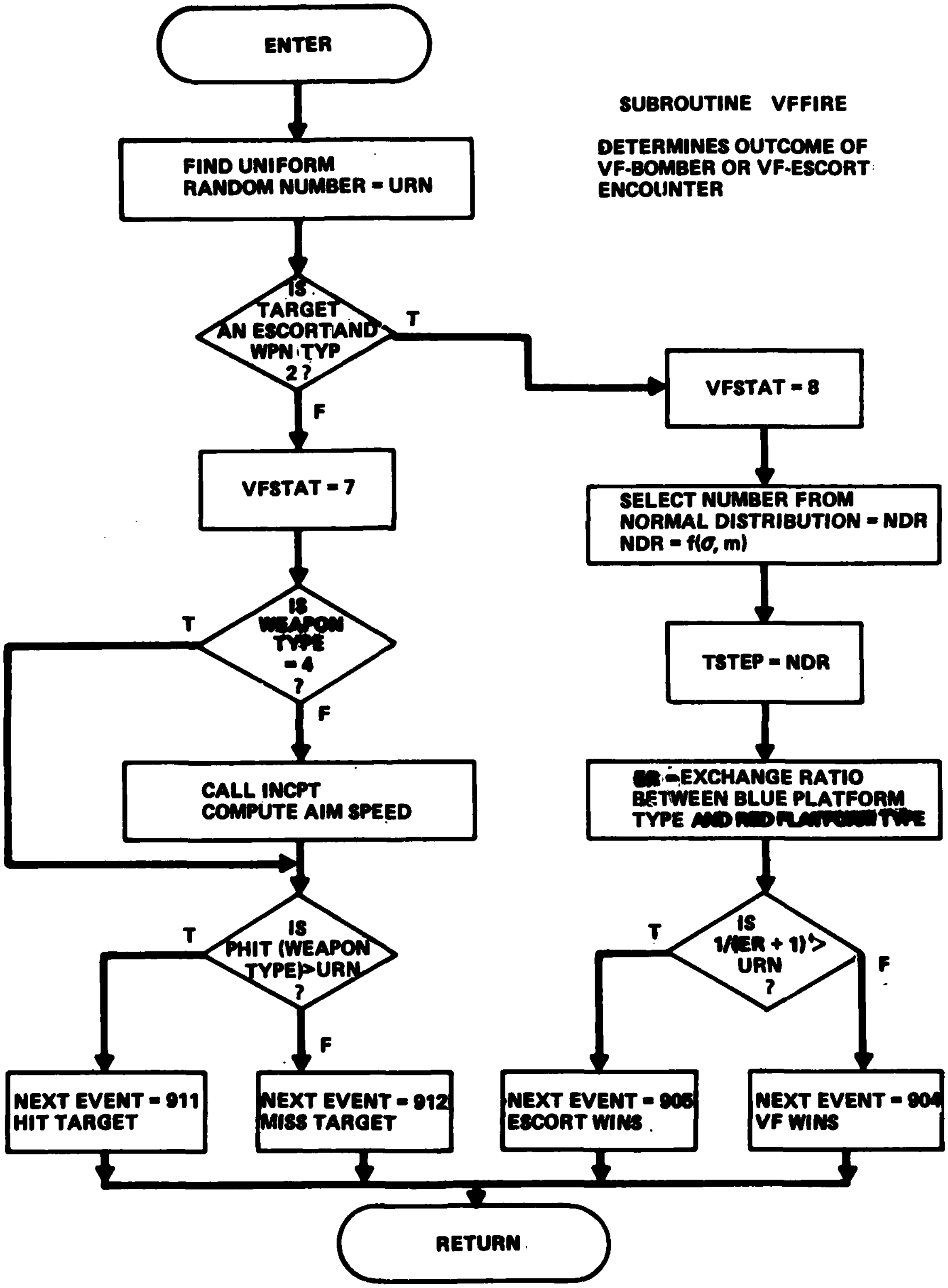


Figure 8-7. Subroutine VFFIRE

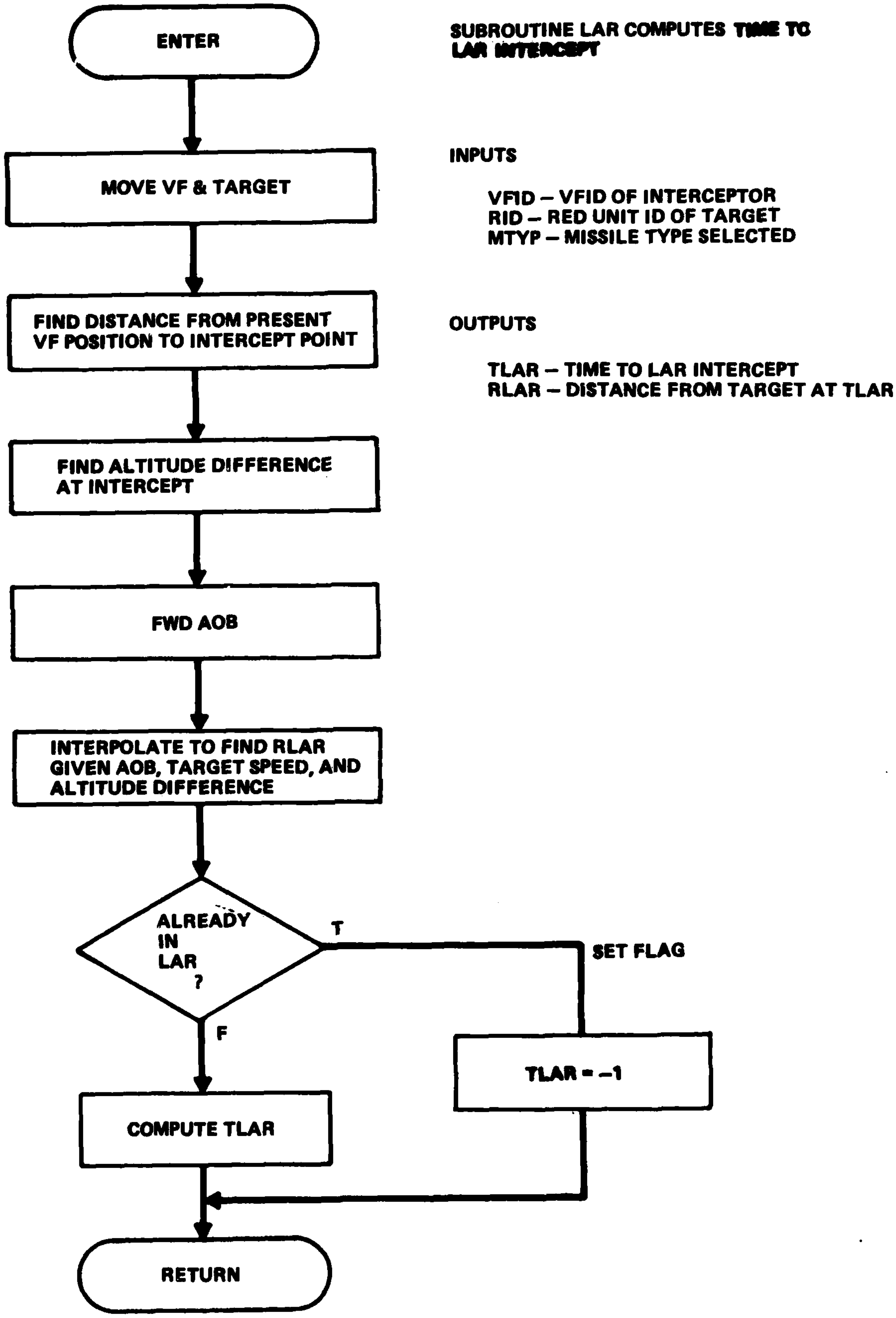


Figure 8-8. Subroutine LAR

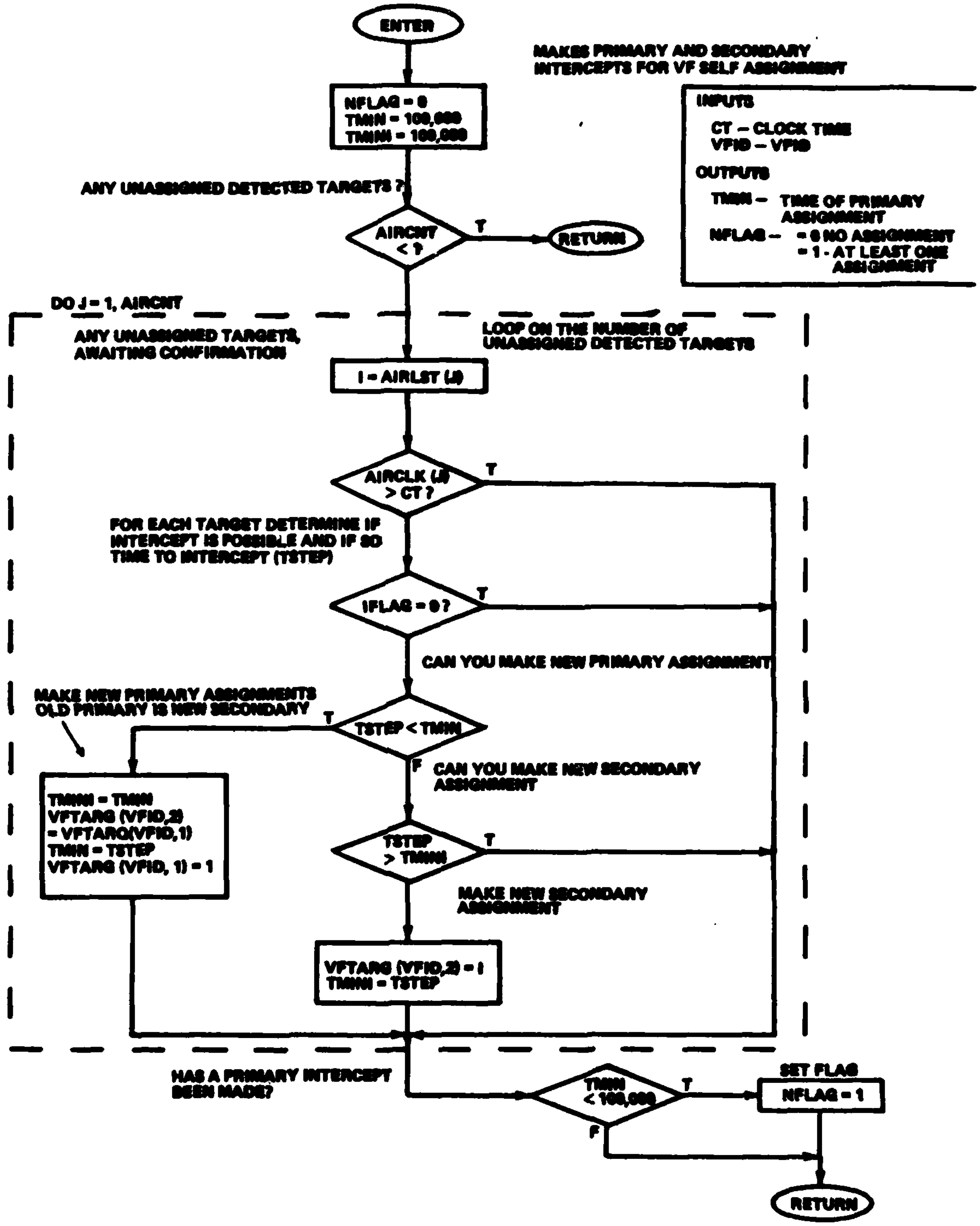


Figure 8-9. Subroutine SELFAS

9. SAM SHIP MODULE

9.1 BASIC SAM LOGIC

The SAM Ship Module simulates those ship functions that process detected targets and Command Center messages, make decisions, and fire SAMs. The targets are either ASMs or enemy aircraft that venture closer to the defended point than the minimum range for interceptors. For purposes of the simulation, the SAM ship is organized into three functions - fire control, launcher, and missile guidance - each of which can have multiple channels. The fire control model includes the time delay for search and acquisition by the fire control radar and the time to develop a target track. The launcher model includes loading, slewing, and launching the SAMs. The guidance model represents the means of guiding the SAM from the ship to the target.

Three generic types of SAMs have been identified for the simulation:

SAMTYP 1 Command All-The-Way
SAMTYP 2 Home All-The-Way
SAMTYP 3 Mid-Course Guidance and Terminal Homing

At this time the first two are treated identically in the simulation. Also for SAMTYP 1 and 2 it is possible to combine the Guidance Channels with the Fire Control Channels. Figure 9-1 shows the functional flow for a ship system using SAMTYP 1 or 2 missiles. The GPSS seizes a Fire Control Channel as soon as one is available after it is required. After a delay of TLOK, the system is ready to slew the launcher if one is loaded and available. After a delay of TSLEW the SAM is launched and the launcher is released to be reloaded. The SAM flies until it reaches the intercept point at time THIT. An evaluation period, TEVAL, follows after which the Fire Control Channel is released and made available for reuse. If a two-round salvo is fired, the inter-launch delay, TDUO, is inserted. A Fire Control Channel can be preempted for a higher priority target up until the SAM is launched.

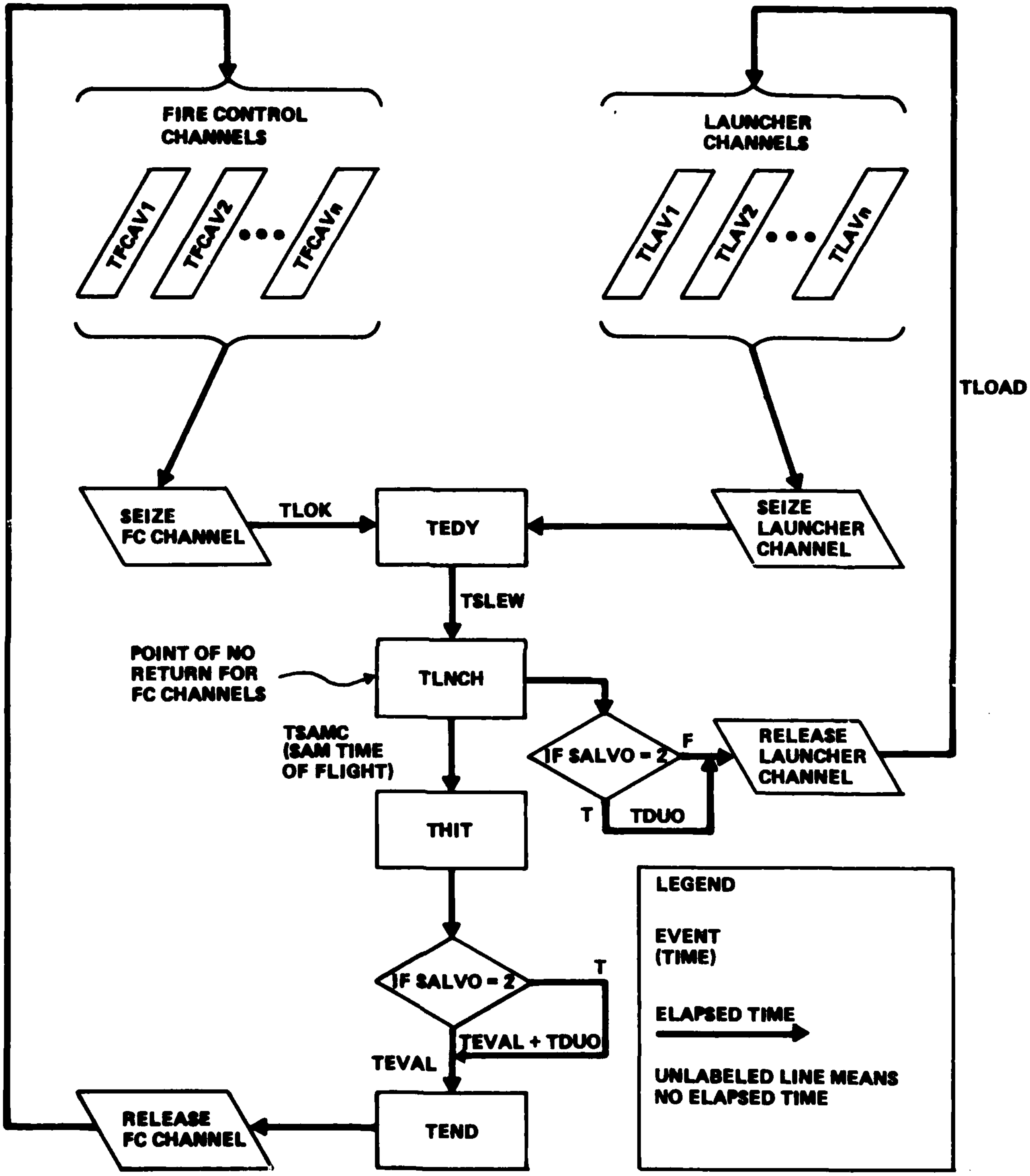


Figure 9-1. Functional Flow with SAMTYP 1 or 2 (Command all-the-Way or Home-all-the-Way)

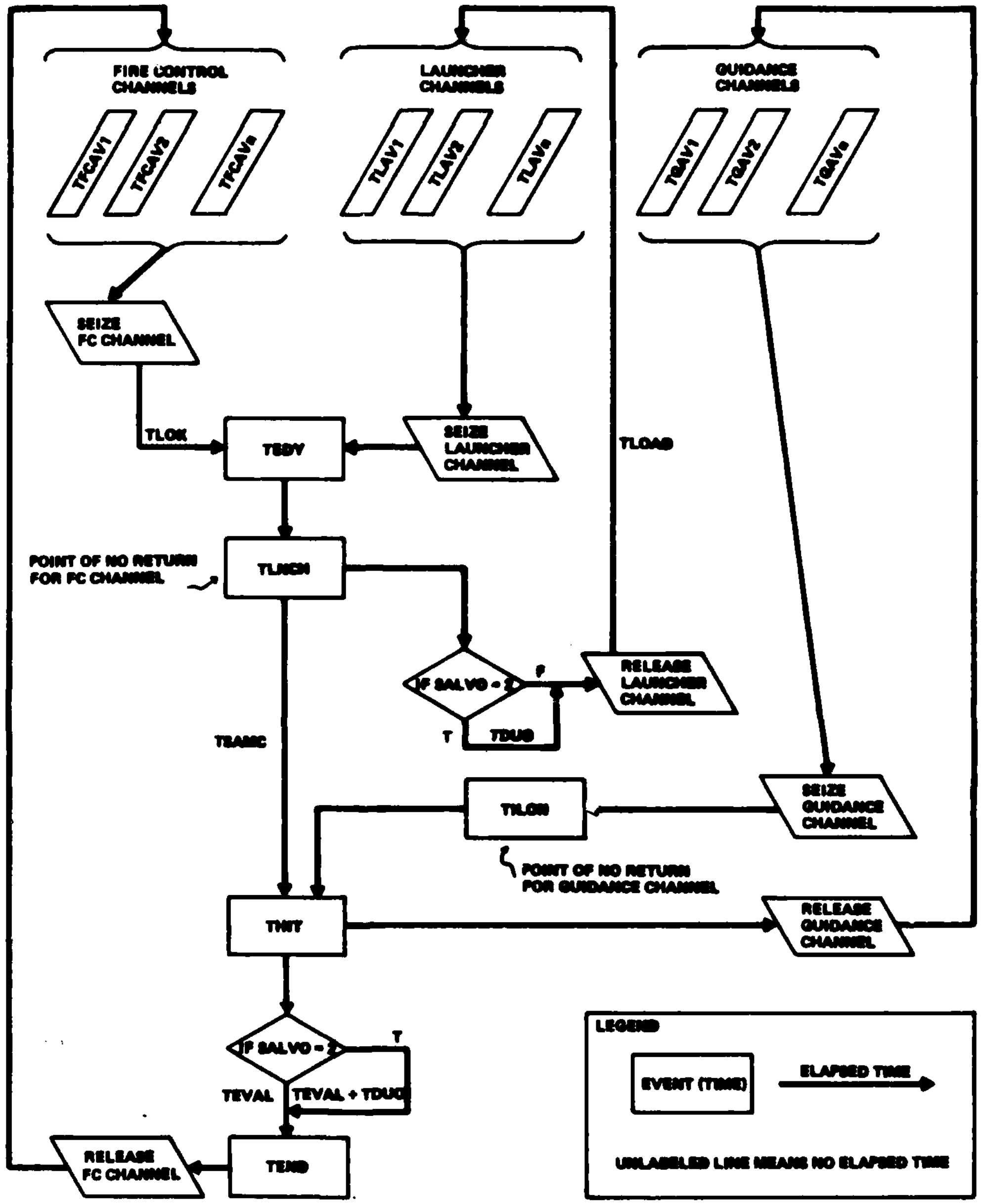
The SAMTYP 3 missile requires that the Guidance Channel, which represents the illuminator, be modeled as a separate element as shown in Figure 9-2. The Guidance Channel must be available during the time TILL before the intercept. It is released after the intercept. The more detailed modeling of these elements will be described in the following sections.

The SAM Ship event diagram in Figure 9-3 is closely related to Figures 9-1 and 9-2, but is more GPSS-oriented. Figure 9-3 covers all three SAM types. Also shown are the Help blocks where the FORTRAN routines are used. It is through these Help blocks that the workings of the simulation will be described.

Before going on to the details of the Help blocks, the relationship of the SAM Ship module to the Command Center should be developed. Three types of coordination are modeled, they are:

CDOCT = 1 Command Center coordination
= 2 Ship Self-Assignment sectors
= 3 No coordination

One of these types is selected for each ship or the entire force by the user. With no coordination, each ship may fire at any target it desires without regard to what other ships are doing. The Command Center is not directing the battle and there may be considerable overkill and unengaged targets. With ship sectors, CDOC=2, the ships also operate independently of the Command Center, but fire only at targets that are in their assigned sectors. Sectors are defined for each ship by the true bearing of the CCW and CW sector limits. Figure 9-4 is an example of sector allocation. Function TBRG (Figure 9-5) is applied to each potential target to evaluate its self-assignability. This type of doctrine reduces the number of unengaged targets and overkills and thus makes better use of the SAMs. Again the Command Center is not coordinating the battle. Note the sectors can be defined to overlap one another. With Command Center coordination, CDOCT=1, the Command Center attempts to assign targets to individual ships in a more optimal fashion. To implement Command Center coordination, the concept of target priorities is introduced. Three priorities are defined:



Fiugre 9-2. Functional Flow with SAMTYP 3 (Mid-Course Guidance)

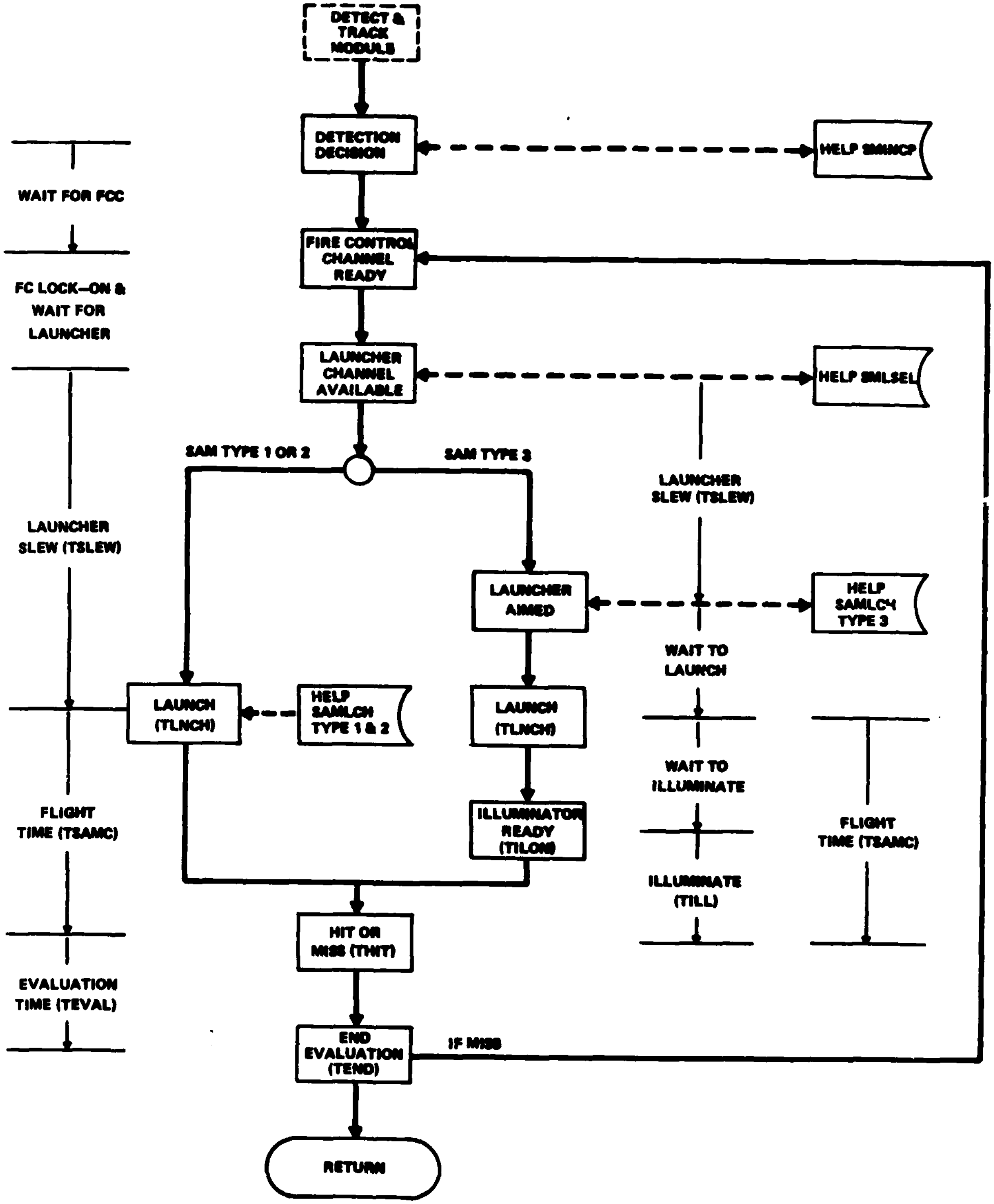


Figure 9-3. SAM Ship Events

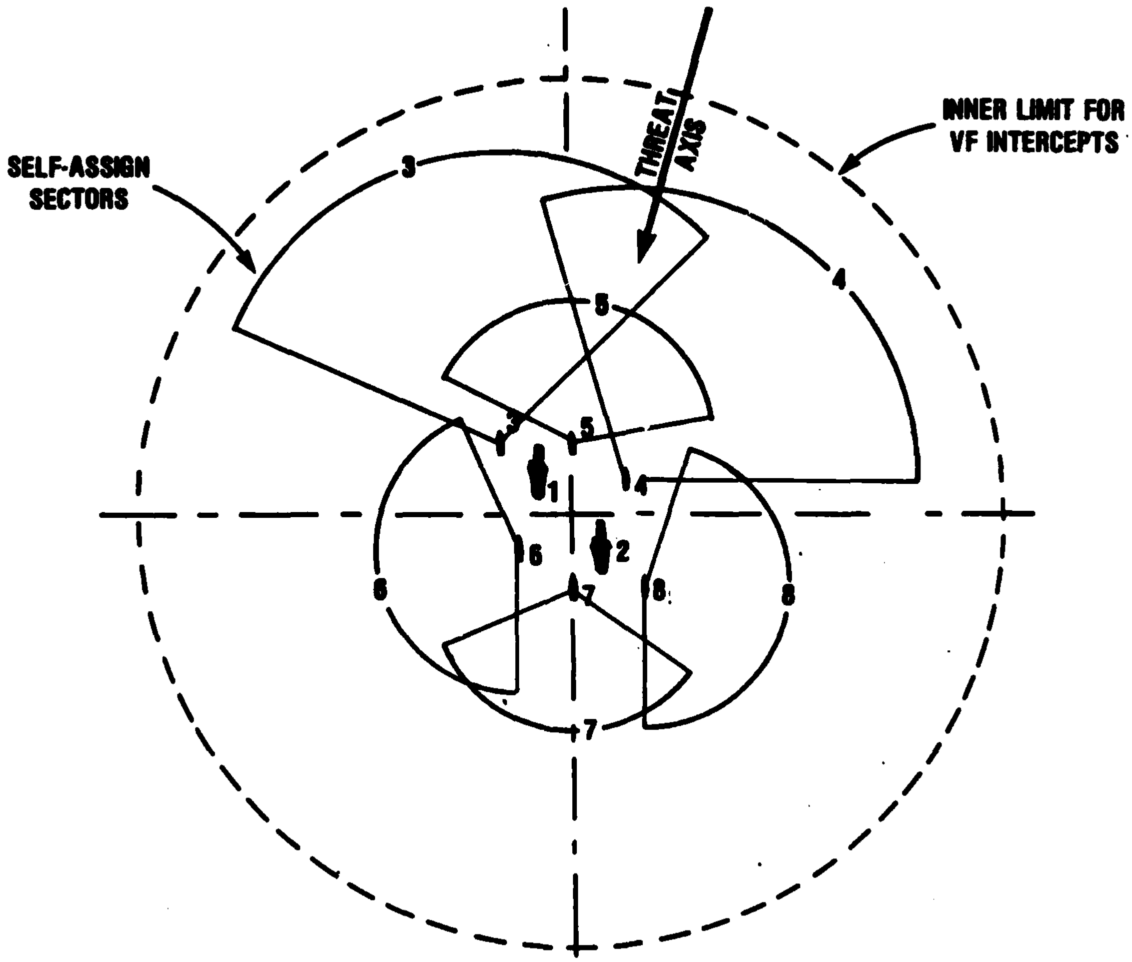


Figure 9-4. SAM Ship Sectors

FUNCTION TBRG(X,Y)

TBRG returns the true bearing from an observer to a point that is X units East and Y units North of the observer's position. Result is in radians clockwise of North, in the range zero to 2π. Arguments can be any real numbers. Defaults to zero for (0., 0.) input.

TBRG also gives true heading for VX and VY velocity components.

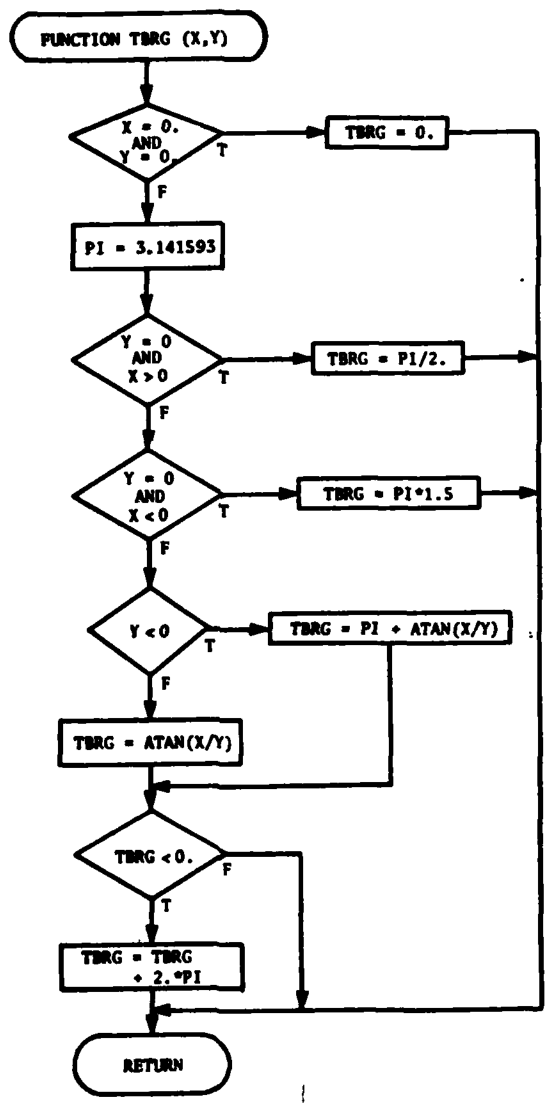


Figure 9-5. Function TBRG

Priority = 1 Threat to ownship
= 2 Command Center assignments
= 3 Ownship self assignments

As each ship makes detections it makes self assignments to shoot at the targets. With Command Center coordination the ships will use sectors as in CDOCT=2. These self assignments are reported to the Command Center which has the power to negate them. The Command Center may make assignments to the individual ships, and these assignments would have a higher priority than a self assignment that had not been reviewed by the Command Center. The highest priority target is one that is perceived as an immediate threat to own ship. Priority 1 targets not only go ahead of Command Center assignments, but they may also preempt or interrupt SAMs about to be launched against lower priority targets. Priority 1 targets may not be preempted. These concepts will be further developed in the following sections and in the description of the Command Center.

Assignments from the Command Center can be rejected by the SAM ship. At the time the command is received, subroutine TGTREJ is called to test if the target is being tracked or if the ship has NTDS; if the ship has more SAMs; and if at least one fire control channel and one launcher channel is up. If any of these tests fail, the assignment is immediately rejected. When the ship actually starts preparations to fire, the target is rejected at any point that the ship determines it cannot get a needed resource in time to hit the target inside the SAM envelope.

9.2 SAM INTERCEPT

The first FORTRAN routine to act on detected targets is SMINCP. This routine determines if the target can be fired on by the ship and establishes the priority. SMINCP is diagrammed in Figure 9-7. Some SAMs must be on the ship for processing to proceed. Next the closest point of approach (CPA) to own ship is computed. This and some of the parameters to be discussed are shown in Figure 9-8. Targets that do not come within the cross range of the SAM are not considered further.

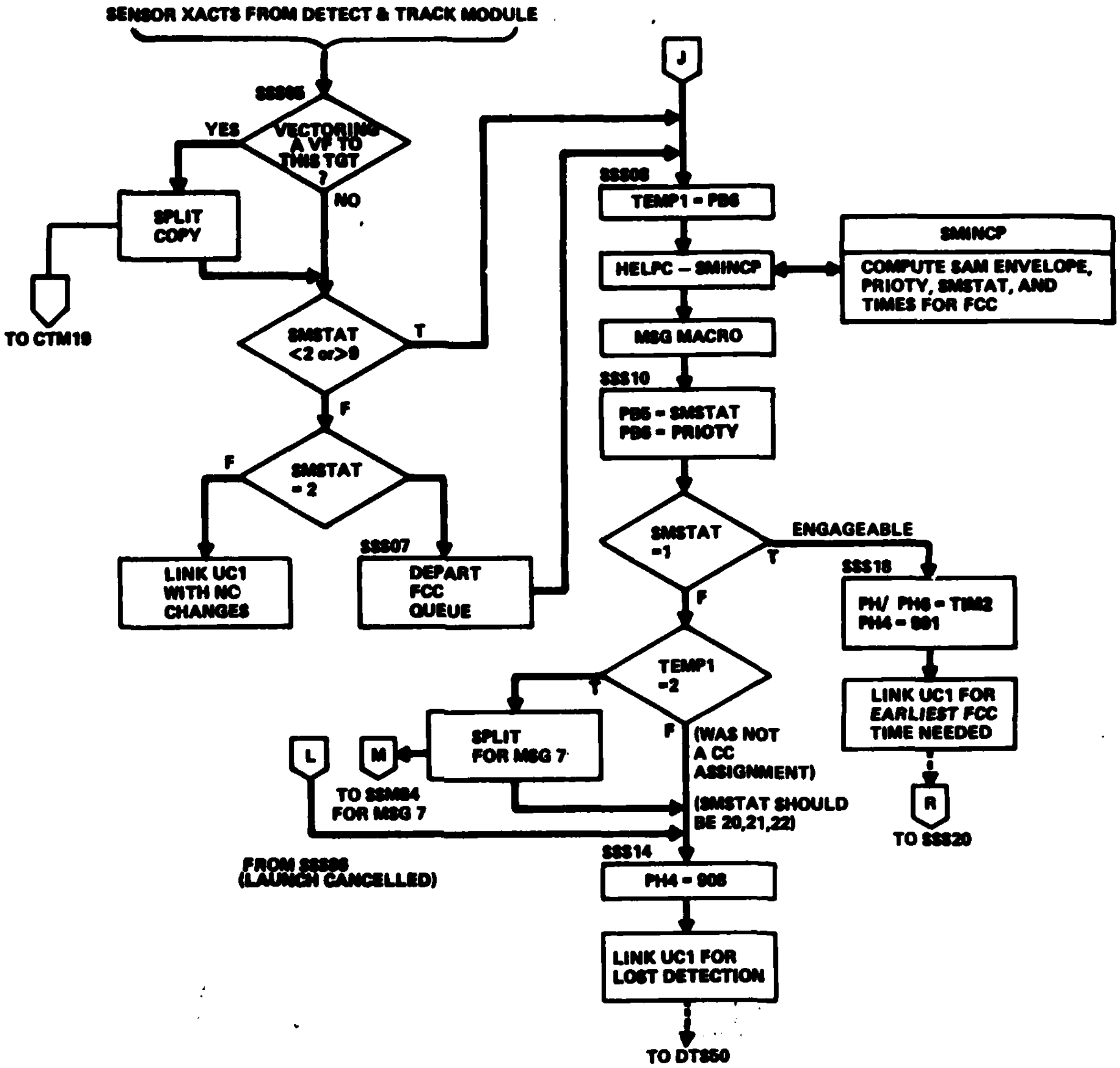


Figure 9-6. SAM Ship GPSS Module (Page 1 of 7)

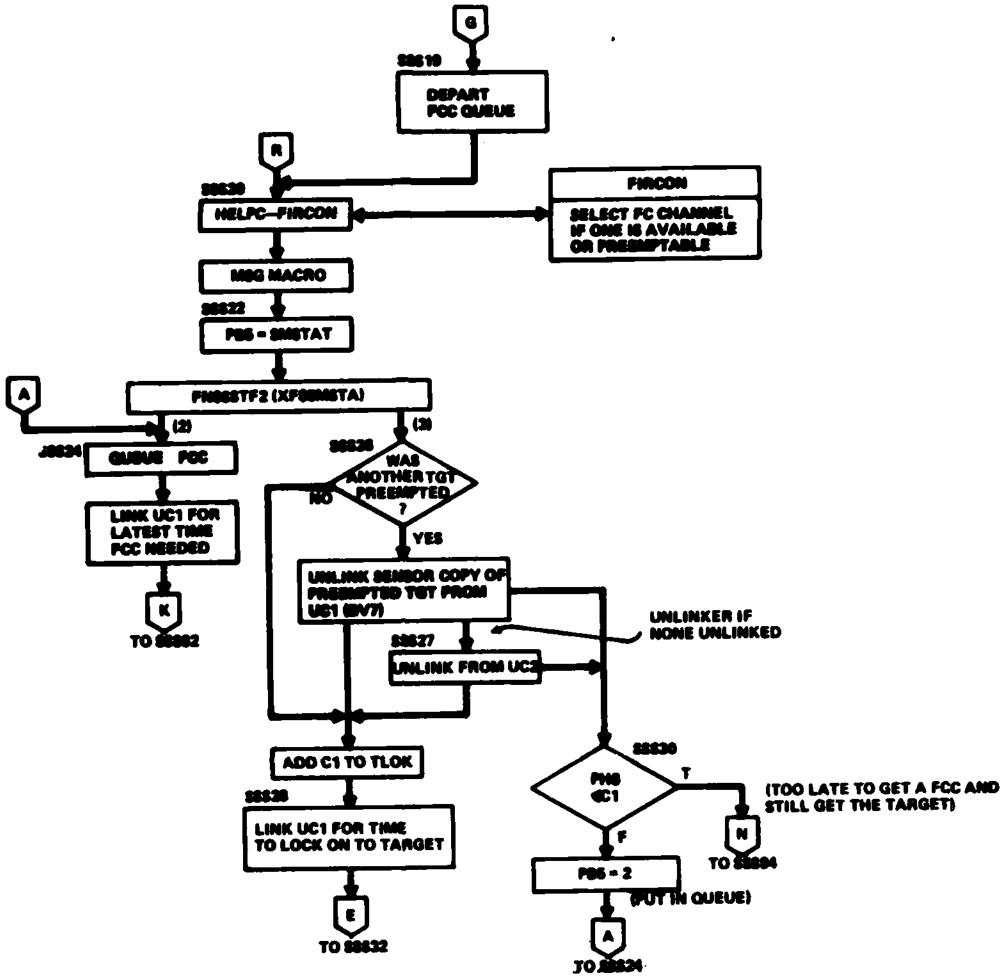


Figure 9-6. SAM Ship GPSS Module (Page 2 of 7)

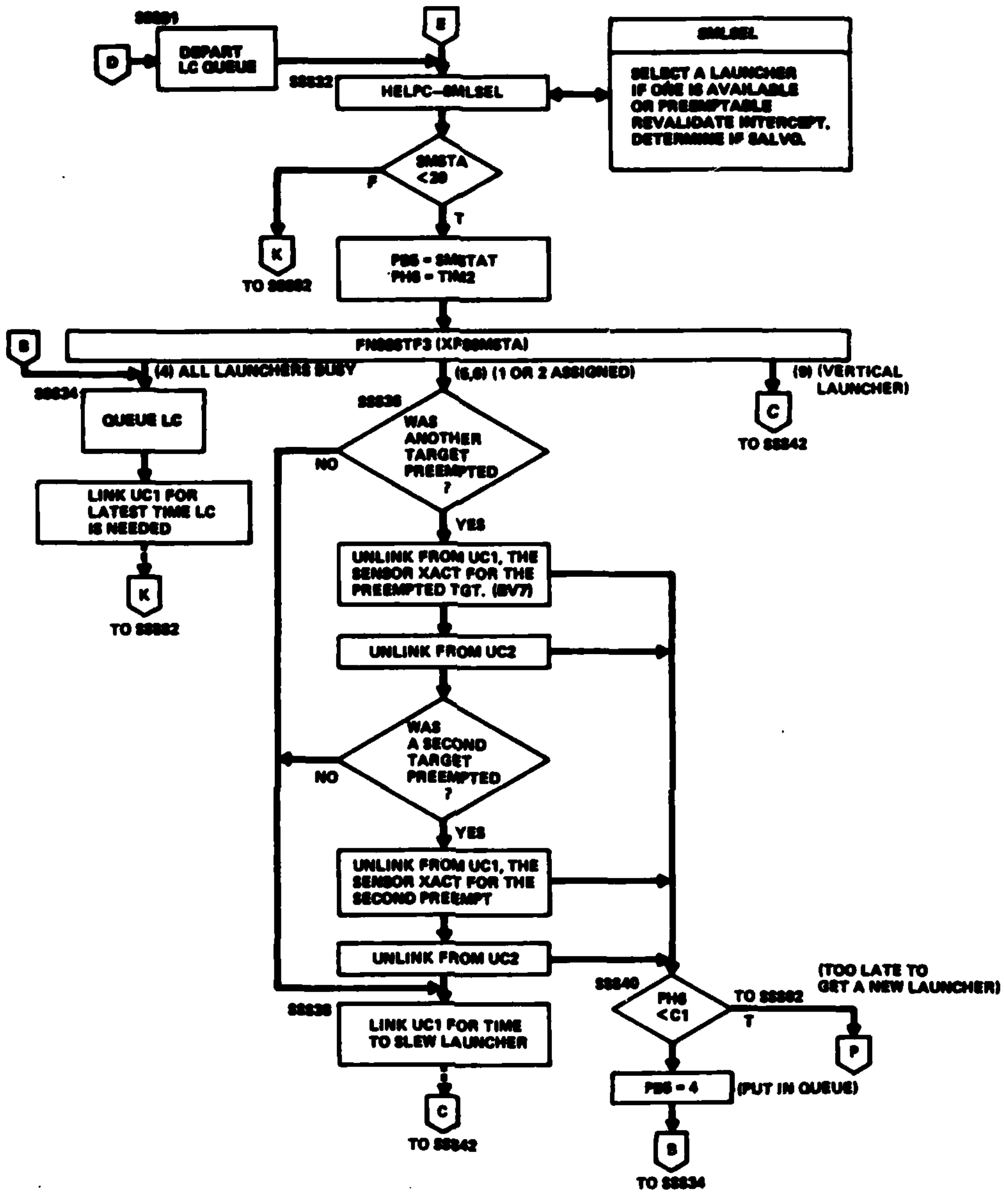


Figure 9-6. SAM Ship GPSS Module (Page 3 of 7)

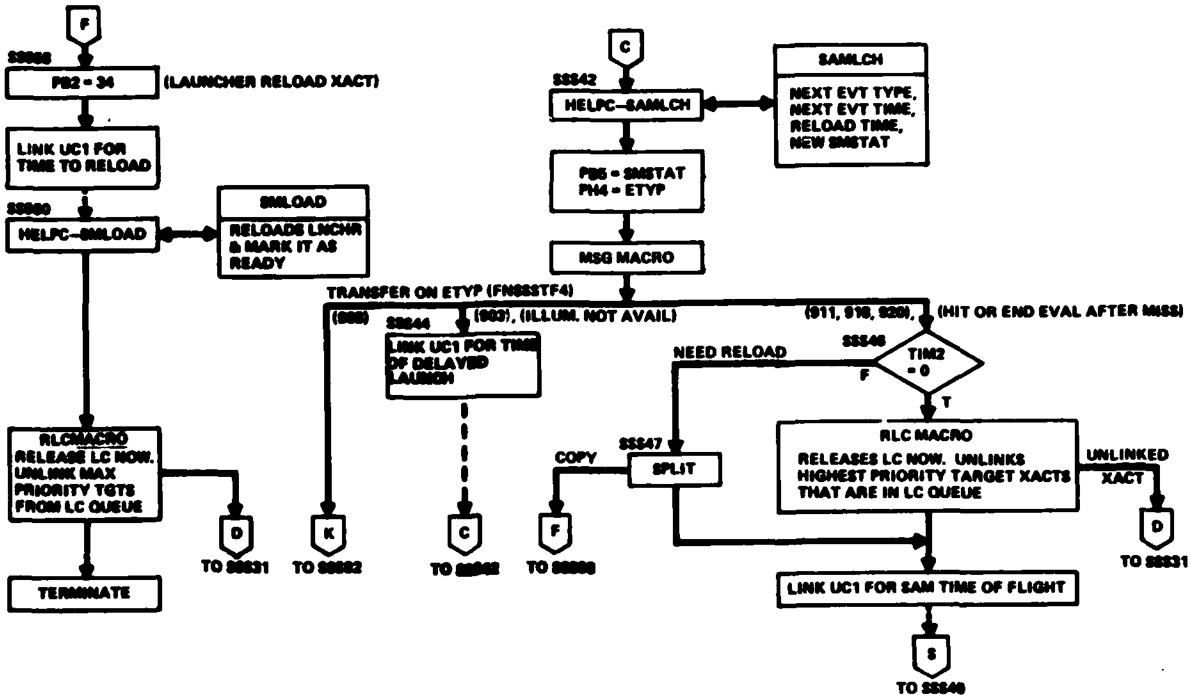


Figure 9-6. SAM Ship GPSS Module (Page 4 of 7)

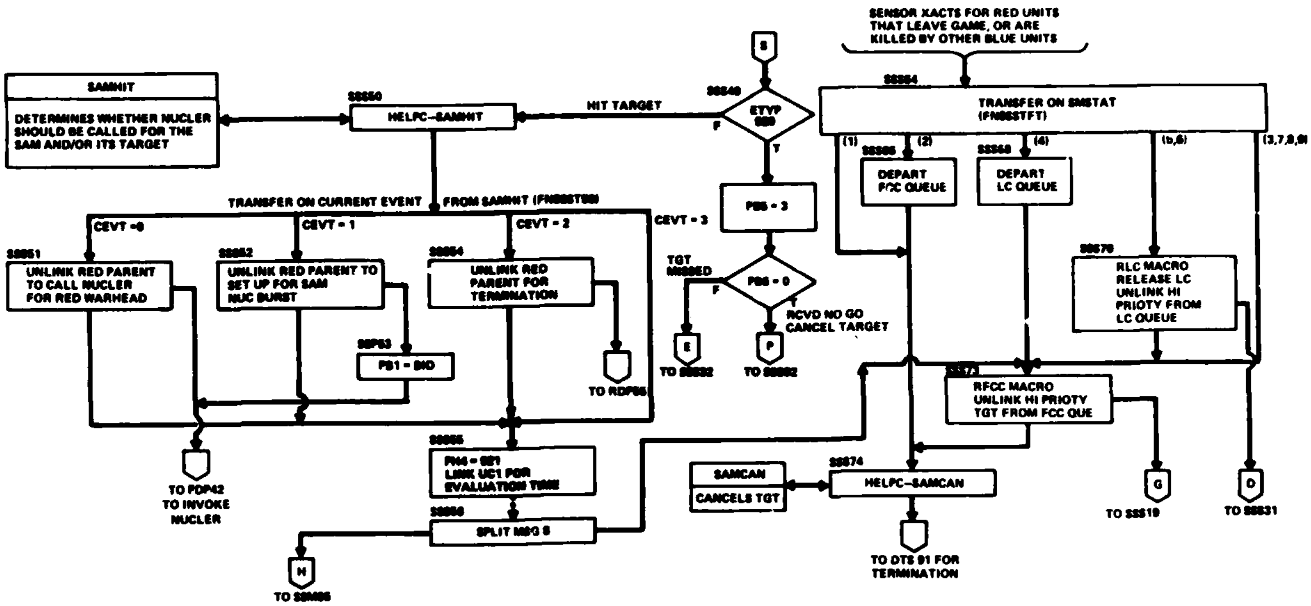


Figure 9-6. SAM Ship GPSS Module (Page 5 of 7)

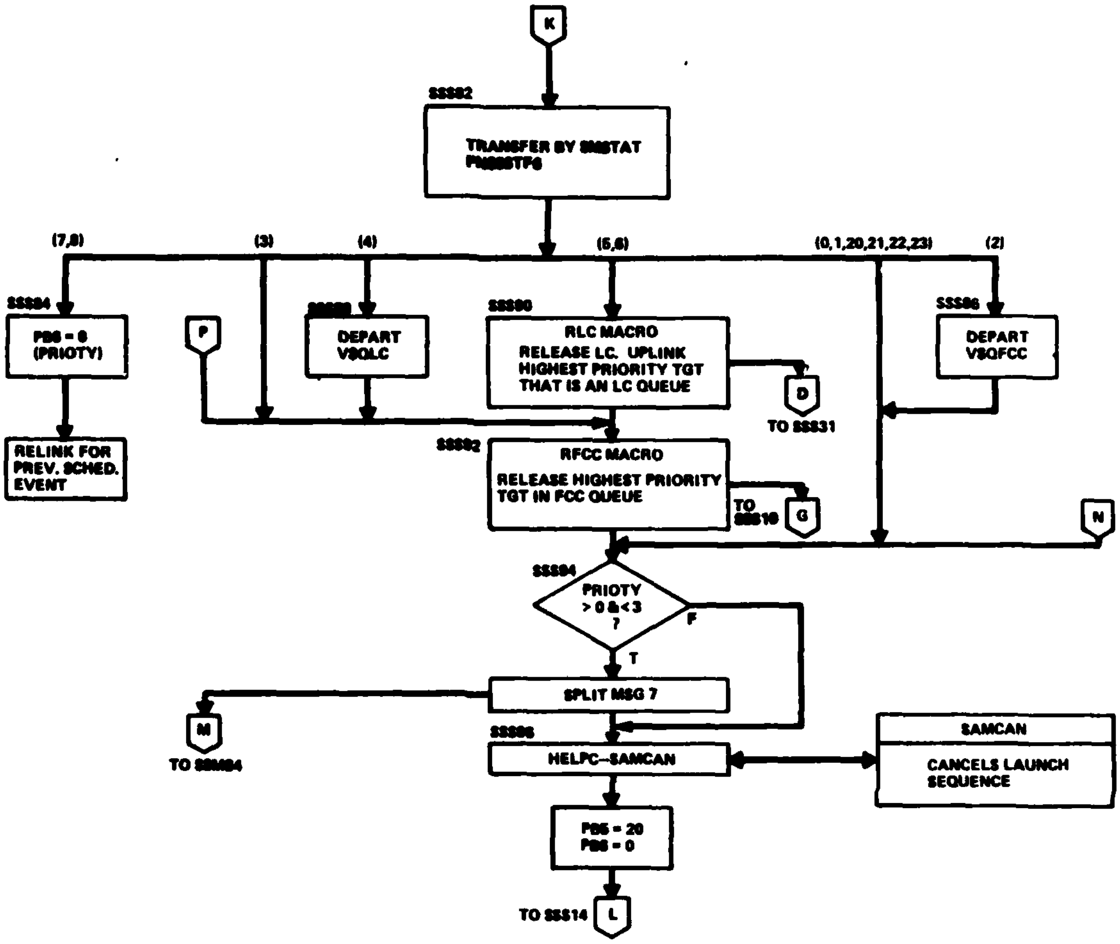


Figure 9-6. SAM Ship GPSS Module (Page 6 of 7)

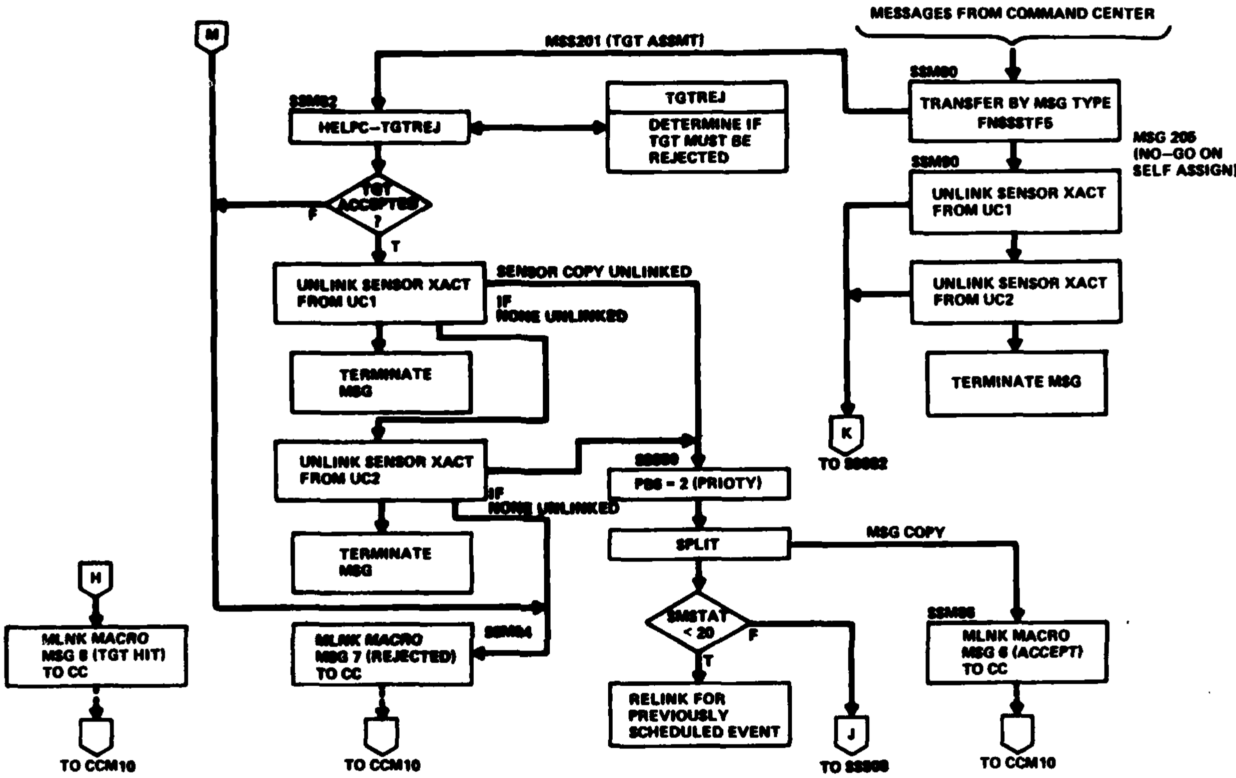


Figure 9-6. SAM Ship GPSS Module (Page 7 of 7)

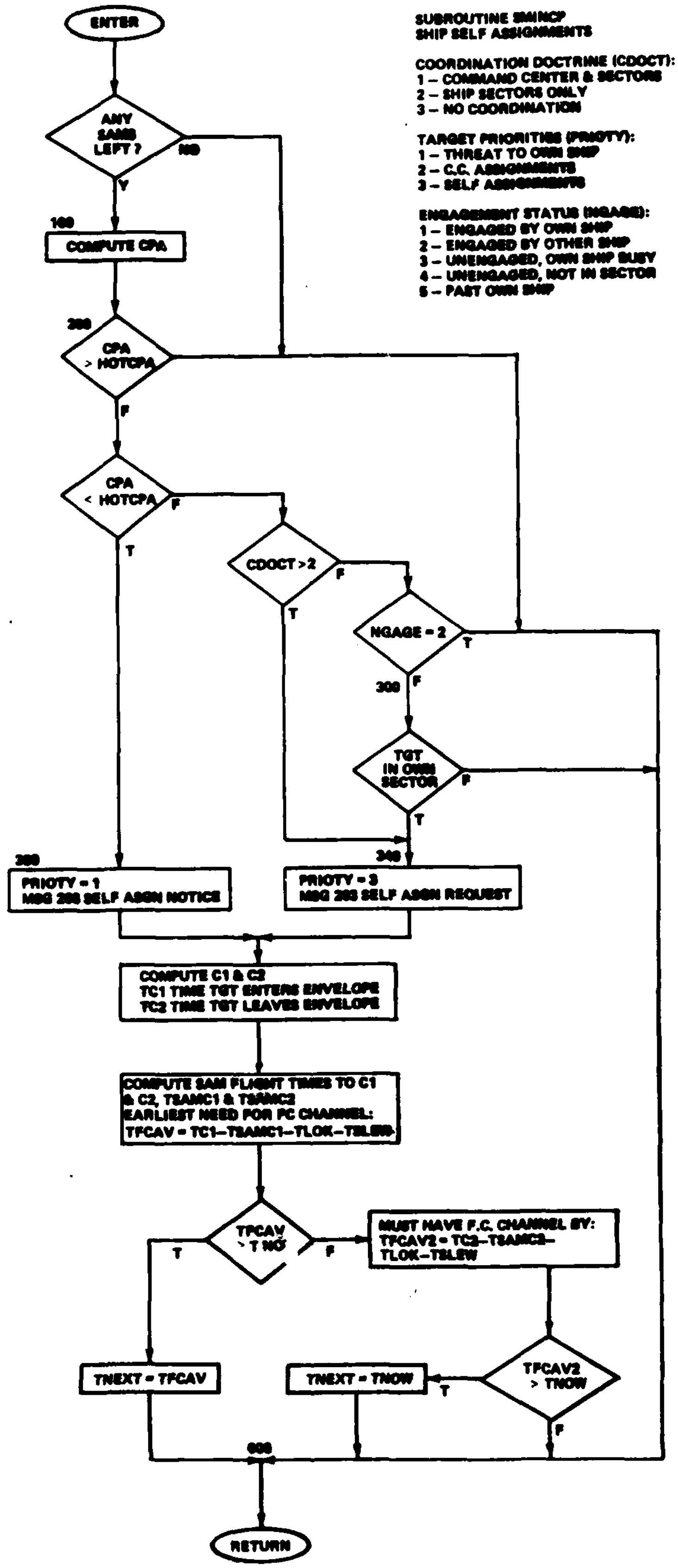


Figure 9-7. Subroutine SMINCP

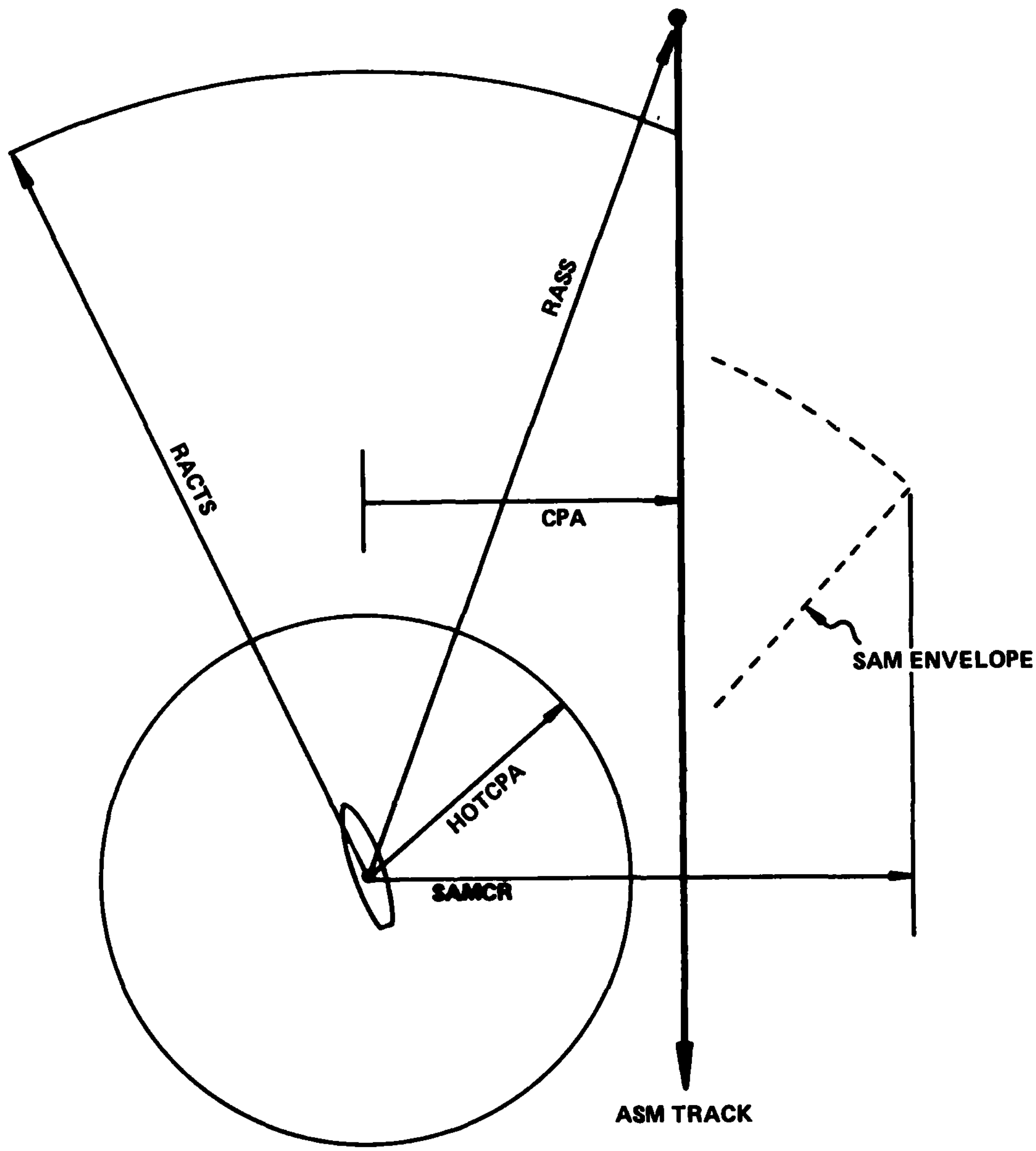


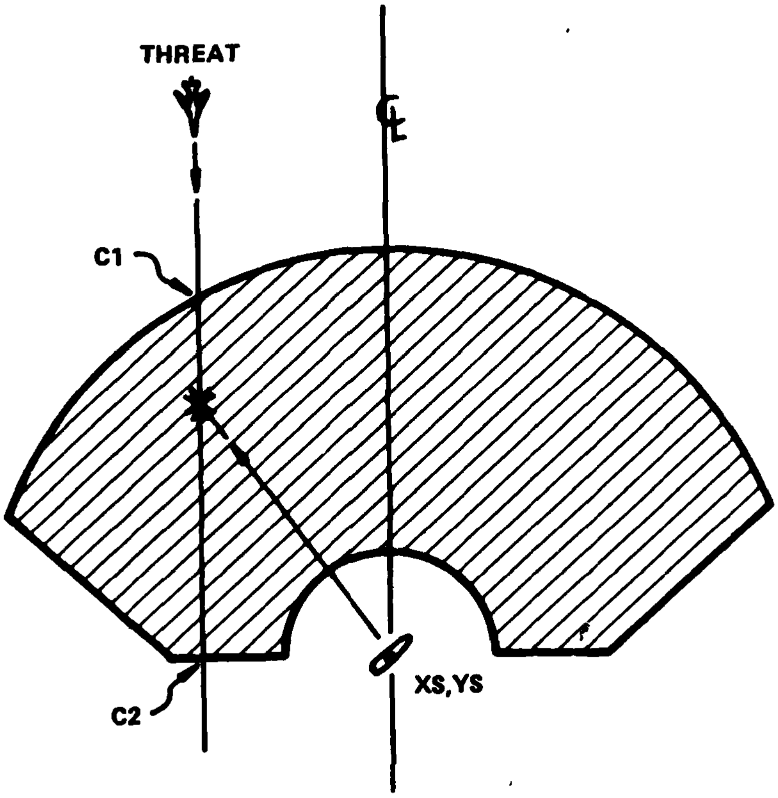
Figure 9-8. SAM Ship Decision Geometry

A ship action range, RACTS, is defined by the user. This is a range beyond which the ship will not process a target. RACTS is intended to prevent the ship from committing resources long before they can be employed. This problem could result from the radar detection ranges greatly exceeding the SAM range. Once the target reaches the action range, the processing continues and targets that will come within the HOTCPA are labeled priority 1 (PRIOTY=1) and a self-assign notice is issued to inform the Command Center of the intent to fire. Targets that do not threaten own ship will be assigned priority 3 status unless and until the Command Center assigns it to the ship. If the ship is using coordination doctrines 1 or 2 it will check to see if the target will be in its sector.

After the priorities are set the SAM envelope crossings are determined (see Figure 9-9). From these times, the earliest time the fire control is needed (derived from the first crossing, C1) is computed. If current time is already past that point, then it must be determined if the intercept can be made before it leaves the envelope at C2. In the first case, the time at which a Fire Control Channel should be seized (see Figure 9-1) is established; in the second case, if the intercept can be made, then the Fire Control Channel should be seized as soon as possible. When no intercept is possible because the target is past the SAM envelope or beyond its cross range capability, the target is marked as not engageable by this ship. No further processing is done unless the target changes course.

Figure 9-10 shows the sequence for finding a Fire Control Channel. First choice is an unoccupied one. With Priority 1 targets an occupied channel may be preempted if no unoccupied ones are available

Referring again to Figure 9-3, the next event occurs when a Fire Control Channel becomes available. When it is ready, the lock-on and track time follows until a launcher is available.



Threat crosses SAM envelope at TC1 and TC2, solution of straight line (target track) and truncated superellipse (SAM envelope)

Figure 9-9. SAM Envelope

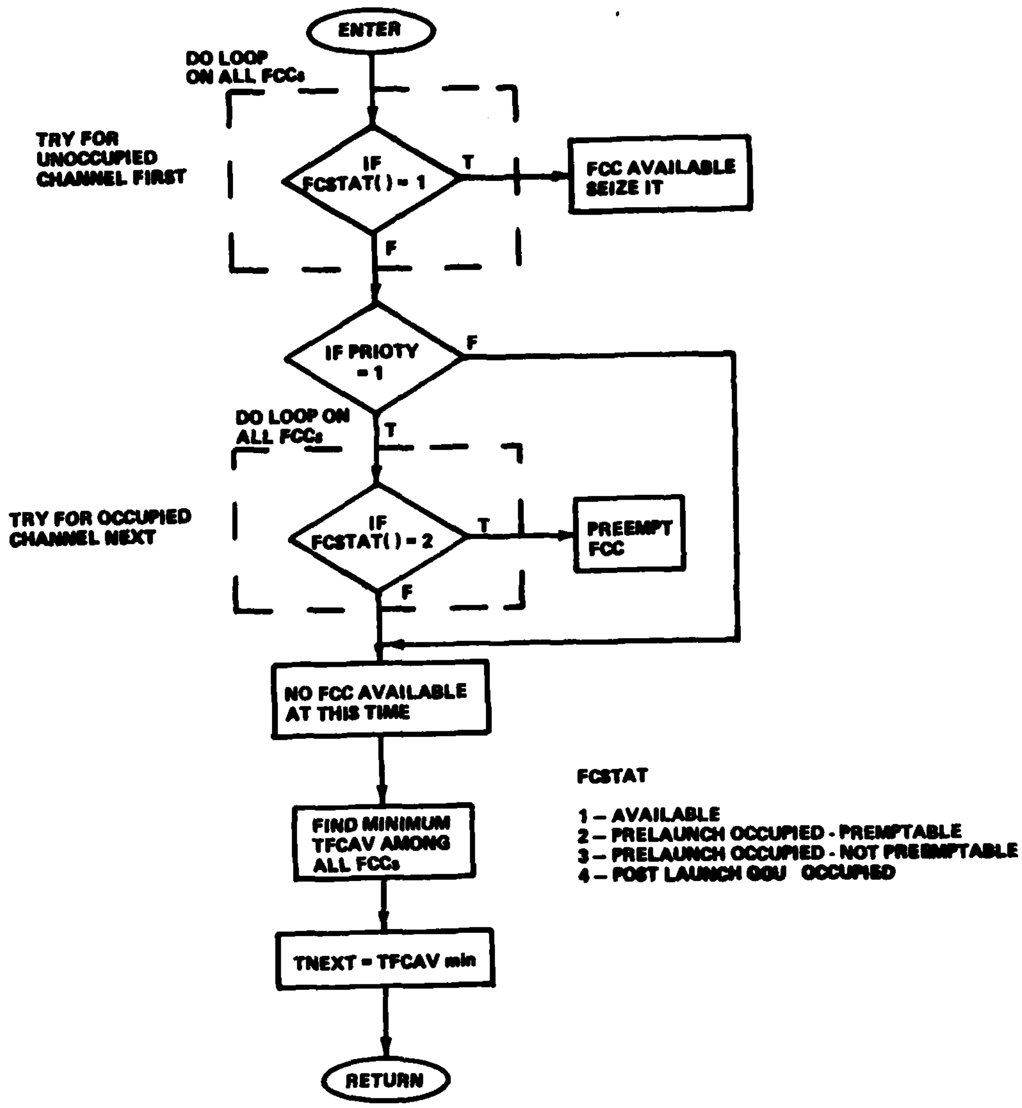


Figure 9-10. Find Fire Control Channel

9.3 SAM LAUNCHER

The functions simulated by this module include removing a SAM from a magazine, mounting on a launcher, aiming the launcher, performing preflight tests and alignments, and launching the SAM. A relatively large number of launcher configurations in the fleet impose the requirement for a flexible launcher model. Three launcher types are modeled - single rail, dual rail, and vertical launching system. The first two currently exist in the fleet, and the third is in development. The vertical launcher consists of fixed cells from which the SAMs are launched without being physically aimed. A brief survey of the fleet indicates ships with configurations as follows:

- o one single rail
- o two single rail
- o one dual rail
- o two dual rail
- o one single rail and two dual rail

Since the last configuration exists only on one ship, just the first four configurations are modeled, in addition to the vertical launching system.

To implement the launcher functions, several launcher states have been identified. The launcher may be "ready" - loaded with a SAM, "in slew" - SAM loaded and launcher being aimed, and "being loaded" - SAM being removed from the magazine and placed on the launcher. Because one or two SAMs may be on the launcher, that number is incorporated in the state number. The launcher states are as follows:

State Number	Description
11	Ready, one SAM
12	Ready, two SAMs
21	In slew, preemptable, one SAM
22	In slew, preemptable, two SAMs, two assigned
23	In slew, preemptable, two SAMs, one assigned

31	In slew, not preemptable, one SAM
32	In slew, not preemptable, two SAMs, two assigned
33	In slew, not preemptable, two SAMs,
41	Being loaded, single rail
42	Being loaded, dual rail

The vertical launcher does not utilize all these states since it is essentially loaded at all times and no slewing is required. However, delay between launches is considered.

Launches are triggered by launch orders of which there are four.

o single, priority 2 or 3
o salvo of two, priority 2 or 3
o single, priority 1
o salvo of two, priority 1

Priority 1 orders can overrule a previous order and prempt the launches; Priorities 2 and 3 cannot.

The launcher selection logic in subroutine SMLSEL is shown in Figure 9-11. The target priority can be specified prior to selection of the launcher. However, the single or salvo decision cannot be established until the time of launch is known. The single or salvo decision is made when at least one launcher is made available. The criterion for a single or salvo of two launch is whether the estimated intercept range is beyond or within RSALVO. Because the actual intercept range is not yet known, it is necessary to estimate the range at intercept in order to set the SALVO switch. First the time of launch is estimated

TTEMP = TNOW + TSLEW + TWAIT

where:

TWAIT = 0 for SAMTYP = 1 or 2 and is
input by the user for SAMTYP = 3

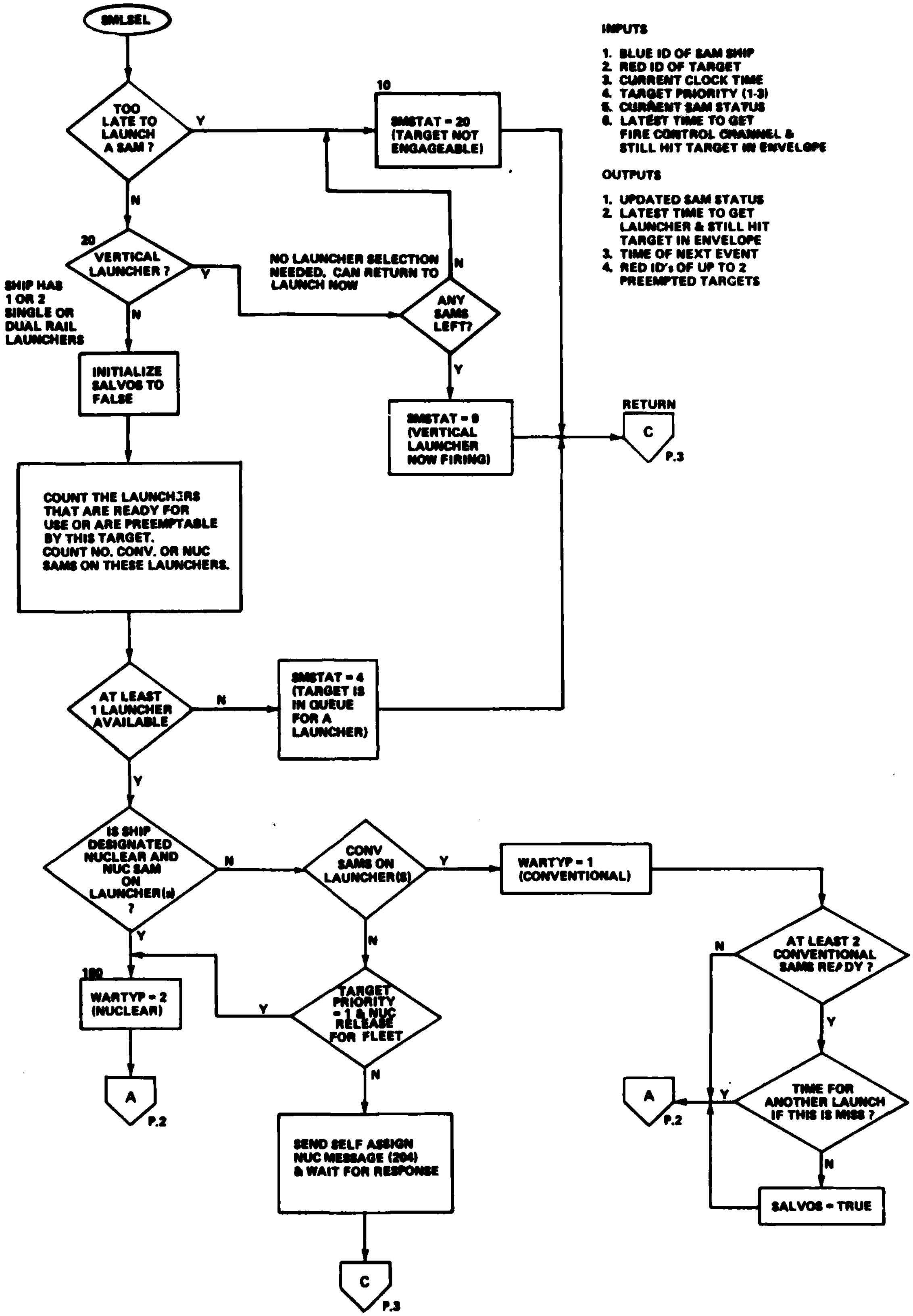


Figure 9-11. Subroutine SMLSEL (FORTRAN)
(Page 1 of 3)

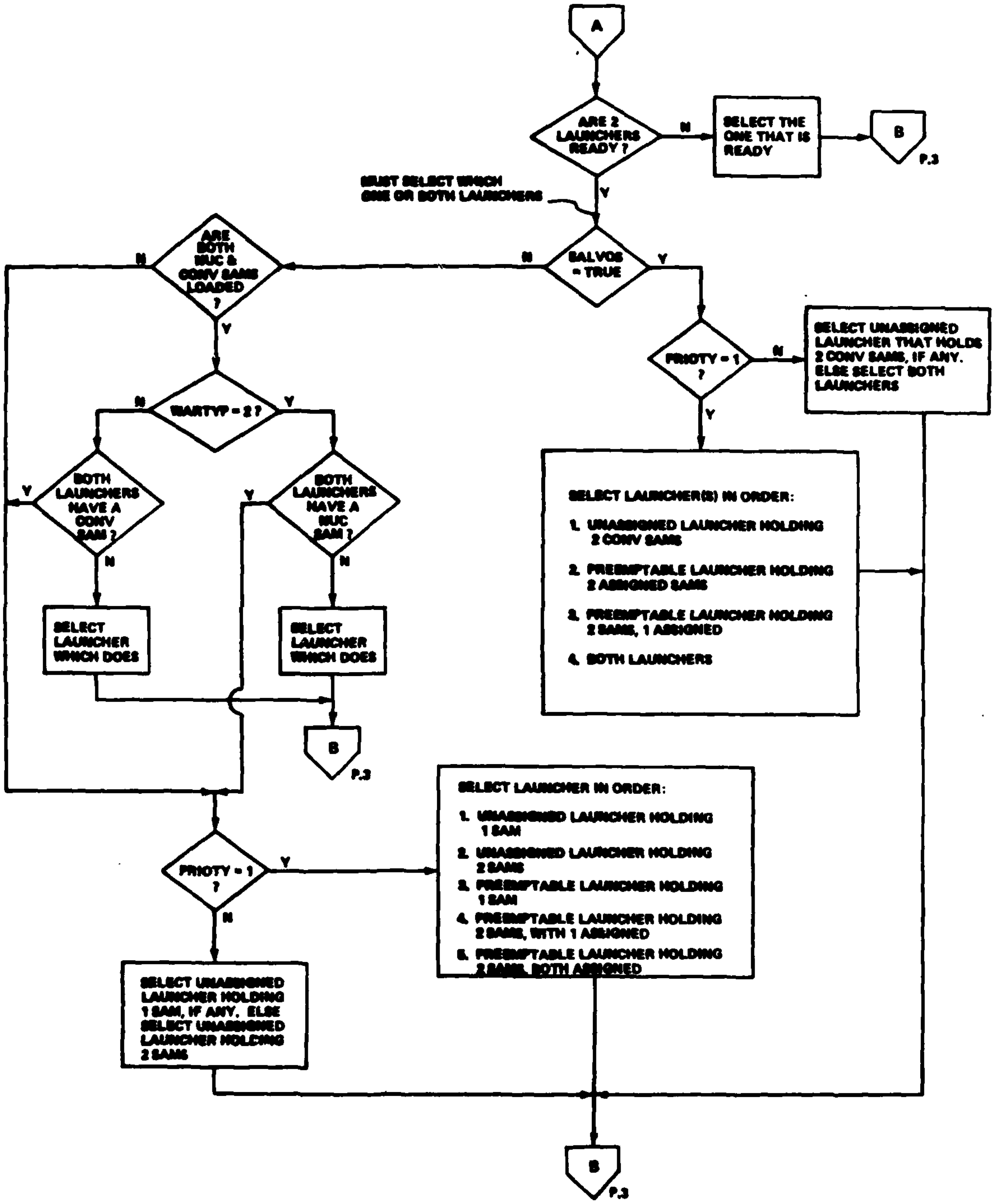


Figure 9-11. Subroutine SMLSEL (Page 2 of 3)

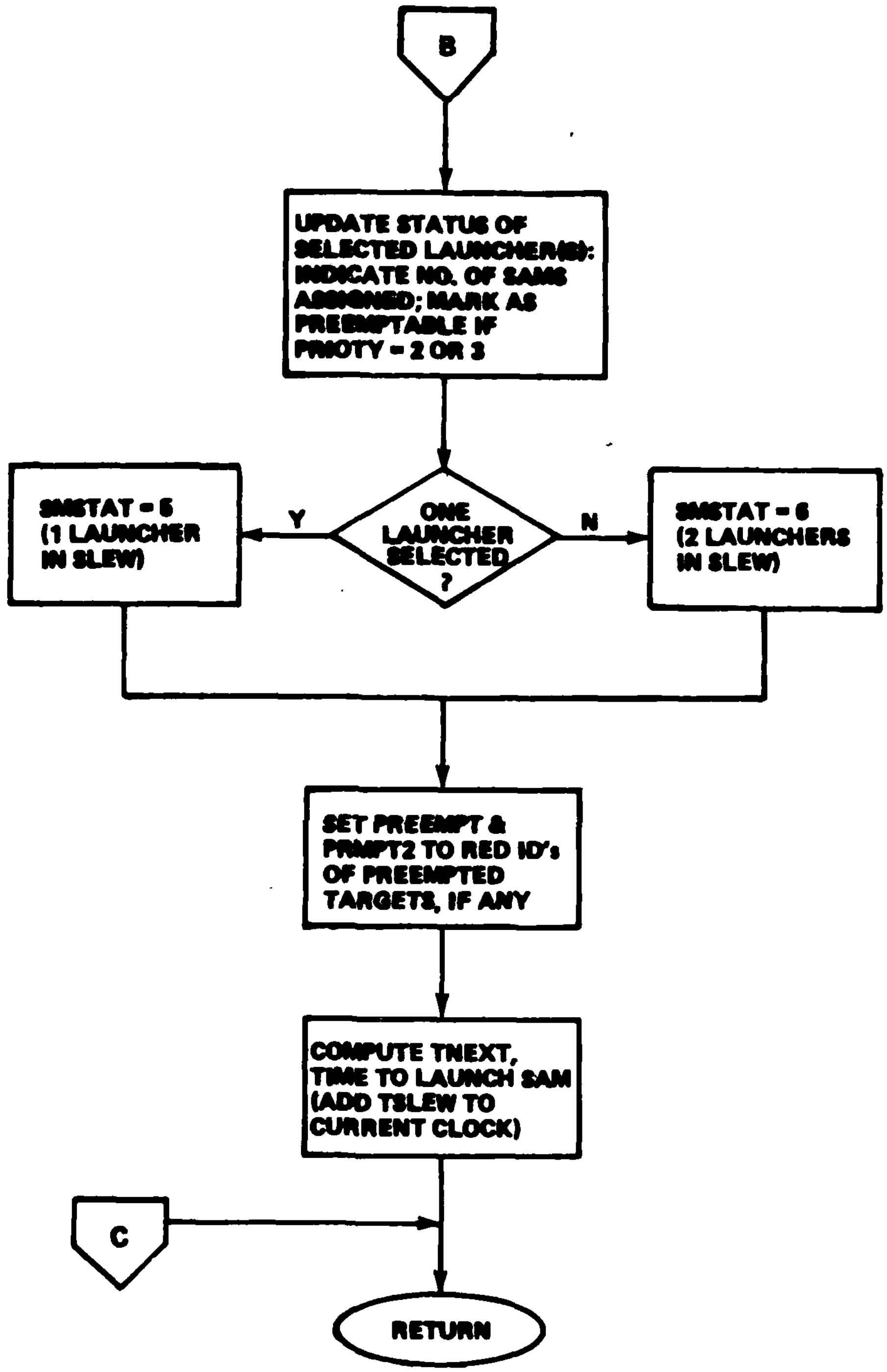


Figure 9-11. Subroutine SMLSEL (Page 3 of 3)

This TWAIT is the time to wait before launching because no illuminator would be available if the launch were to occur immediately. A "typical" value is used to represent what is actually an almost unpredictable function of game conditions. Next the target is temporarily moved to its tentative position at TTEMP and the intercept routine is called for the SAM and the target. The range from ship to target at intercept is then tested against RSALVO and the launch order is complete.

Since the vertical launcher and the rail launcher have vastly different characteristics they are treated separately. The rail launcher requires several steps to make assignments. If there are two launchers on board, the one to use first must be selected. The logic to do this is shown in Figure 9-12. For any given launch order the sequence for testing the launcher states is down the page. If a launcher can be found, then the new states must be set and the length of time in the new state established (see Figure 9-13). If a second launcher is required, e.g., when a salvo of two is ordered on ship with two single rails, the new state and time for the second launcher must be set. A single launch from a launcher with two SAMs is immediately ready with one SAM (State 11) after launch. The length of time spent in the ready states is indefinite, depending on game conditions.

In implementing the just-described model several rules were followed:

- A launcher is reloaded immediately after it is empty. On a dual rail launcher, both SAMs must be launched before reloading.
- For a salvo of two, if two SAMs are not available, fire one and fire the second as soon as possible up until assessment time is complete for the first launch.
- The only targets that preempt are those that threaten own ship.
- The only assigned targets that cannot be preempted are those that threaten own ship.
- A preempt action removes a target that has not been processed to a launch and replaces it with a target that is threatening own ship.

	Launcher States		
Launch Order	Launcher 1	Launcher 2	Launcher Assignment
Salvo, Priority 2 or 3	12		#1
		12	#2
	11		#1
		11	#2
			None Avail.
Salvo, Priority 1	12		#1
		12	#2
	11		#1
		11	#2
	22		#1
		22	#2
	23		#1
		23	#2
	21		#1
		21	#2
			None Avail.
Single, Priority 2 or 3	11		#1
		11	#2
	12		#1
		12	#2
			None Avail.
Single, Priority 1	11		#1
		11	#2
	12		#1
		12	#2
	21		#1
		21	#2
	23		#1
		23	#2
	22		#1
		22	#2
			None Avail.

Figure 9-12. Launcher Selection Logic for Ships with Two Launchers

	NEXT STATE/DURATION					
	LAUNCH ORDERS					
CURRENT STATE	SINGLE PRIOTY = 2 or 3	SINGLE PRIOTY = 1	SALVO PRIOTY = 2 or 3	SALVO PRIOTY = 1	SAM AWAY	LAUNCHER LOADED
11 READY, ONE SAM (SINGLE RI RAIL OR DUAL RAIL WITH ONE SAM)	21 / TSLEW	31 / TSLEW	21* / TSLEW	31* / TSLEW		
12 READY, TWO SAMs (DUAL RAIL)	23 / TSLEW	33 / TSLEW	22 / TSLEW	32 / TSLEW		
21 IN SLEW, PREEMPTABLE, ONE SAM		31 / TSLEW		31* / TSLEW	41 or 42 / TLOAD	
22 IN SLEW, PREEMPTABLE, TWO SAMs, TWO ASSIGNED		33 / TSLEW		32 / TSLEW	42 / TLOAD	
23 IN SLEW, PREEMPTABLE, TWO SAMs, ONE ASSIGNED		33 / TSLEW		32 / TSLEW	11 / IND	
31 IN SLEW, NOT PREEMPTABLE, ONE SAM					41 or 42 / TLOAD	
32 IN SLEW, NOT PREEMPTABLE, TWO SAMs, TWO ASSIGNED					42 / TLOAD	
33 IN SLEW, NOT PREEMPTABLE, TWO SAMs, ONE ASSIGNED					11 / IND	
41 BEING LOADED, SINGLE RAIL						11 / IND
42 BEING LOADED, DUAL RAIL						12 / IND

* SECOND LAUNCH REQUIRED IND – INDEFINITE

Figure 9-13. Launcher State & Time Selection

- A "no-go" from the Command Center on a self-assigned target interrupts the processing of the target up to the point of launch. It does not, however, replace it with another target.

- Selection of a nuclear weapon will utilize single launch orders.

- NNWS may assign certain launchers to ASW, in which case they will be unavailable for AAW.

SAM Envelope Intersection Logic

Subroutine SAMENV computes the time at which enemy aircraft or missiles enter and leave the SAM envelope of a particular ship. To compute these intersection points, the position and speed of the red target are transferred from the X-Y coordinate system to a U-V coordinate system. The V direction is parallel to but opposite in direction to the path of the incoming red target as shown in Figure 9-14. The variable names for the red and blue units are given below:

C1 - X-speed component of red unit
C2 - X-coordinate of red unit
C3 - Y-speed component of red unit
C4 - Y-coordinate of red unit
XS - X-coordinate of blue ship
YS - Y-coordinate of blue ship
θ - Angle that the path of the red unit makes with the x axis (90^o)
D2 - U-coordinate of red unit
D4 - V-coordinate of red unit

The transformation equations from the X-Y to the U-V coordinate system are shown below where S1 through S8 are coefficients based upon the direction of the incoming target.

$$D2 = S1* (C2-XS)*SIN(\theta)+S2*(C4-YS)*COS(\theta)$$
$$D4 = S3* (C2-XS)*COS(\theta)+S4*(C4-YS)*SIN(\theta)$$

The transformation from the U-V back to the X-Y coordinate system is given by:

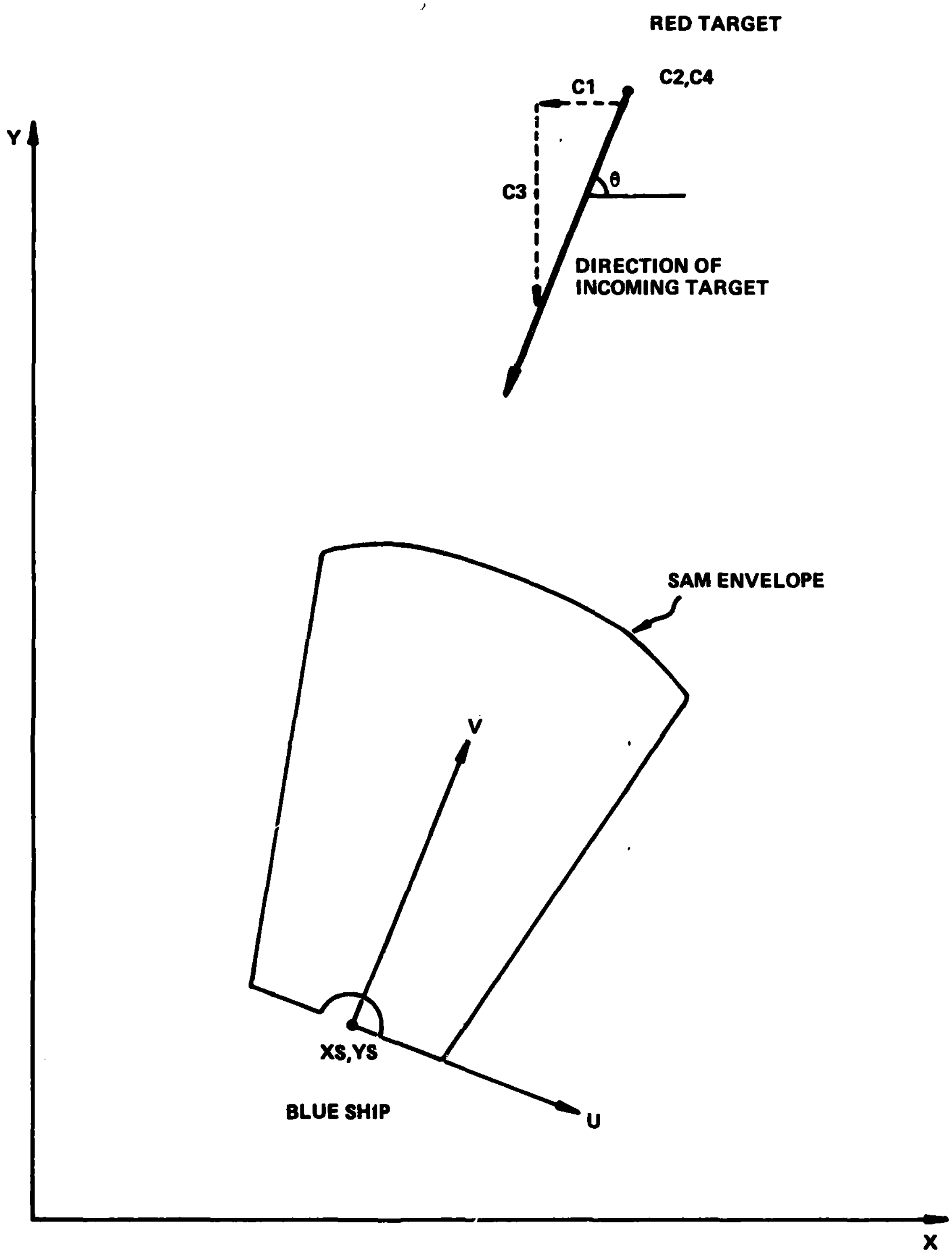


Figure 9-14. Alignment of SAM Envelope to Target

C2=S5*D2*SIN(θ)+S6*D4*COS(θ)+XS

C4=S7*D2*COS(θ)+S8*D4*SIN(θ)+YS

By requiring that the V direction be parallel to the direction of the incoming target, the U coordinate of intercept will be the same whether the target is entering or leaving the envelope. Once D2 (=U1=U2) is calculated, it can be used to calculate the V coordinate of intercept for both intersection points.

The basic SAM envelope configuration is shown in Figure 9-15. The front portion (Zone 1) consists of a segment of a super ellipse given by the equation:

$$\left[\frac{U}{A}\right]^N + \left[\frac{V}{B}\right]^N = C$$

Where A, B, C and N are coefficients that fit the superellipse to the SAM envelope. The rear portion of the envelope consists of a minimum intercept sphere (Zone 2) closest to the ship which connects to a line at V=0. This line (Zone 3) stops at the minimum cross range. The envelope is completed by connecting the minimum cross range at V=0 to the point where the maximum cross range cuts off the superellipse (Zone 4). Thus each SAM envelope can be defined by seven variables

SM1 - SAM maximum cross range

SM2 - SAM minimum cross range

$$\left.\begin{array}{l}\text{SM4 - A}\\ \text{SM5 - B}\\ \text{SM6 - N}\\ \text{SM7 - C}\end{array}\right\}\ \text{Coefficients of superellipse equation}$$

These values will be stored as an Nx7 array where N is the number of different types of SAM envelopes.

Subroutine SAMENV which is called by subroutine SMINCP first calculates D2, the U-coordinate of intercept. Once D2 is known it can be substituted into the superellipse equation to find the V coordinate of the first intercept. The logic then determines which of the three remaining zones the target will intercept and calculates the V coordinate of the

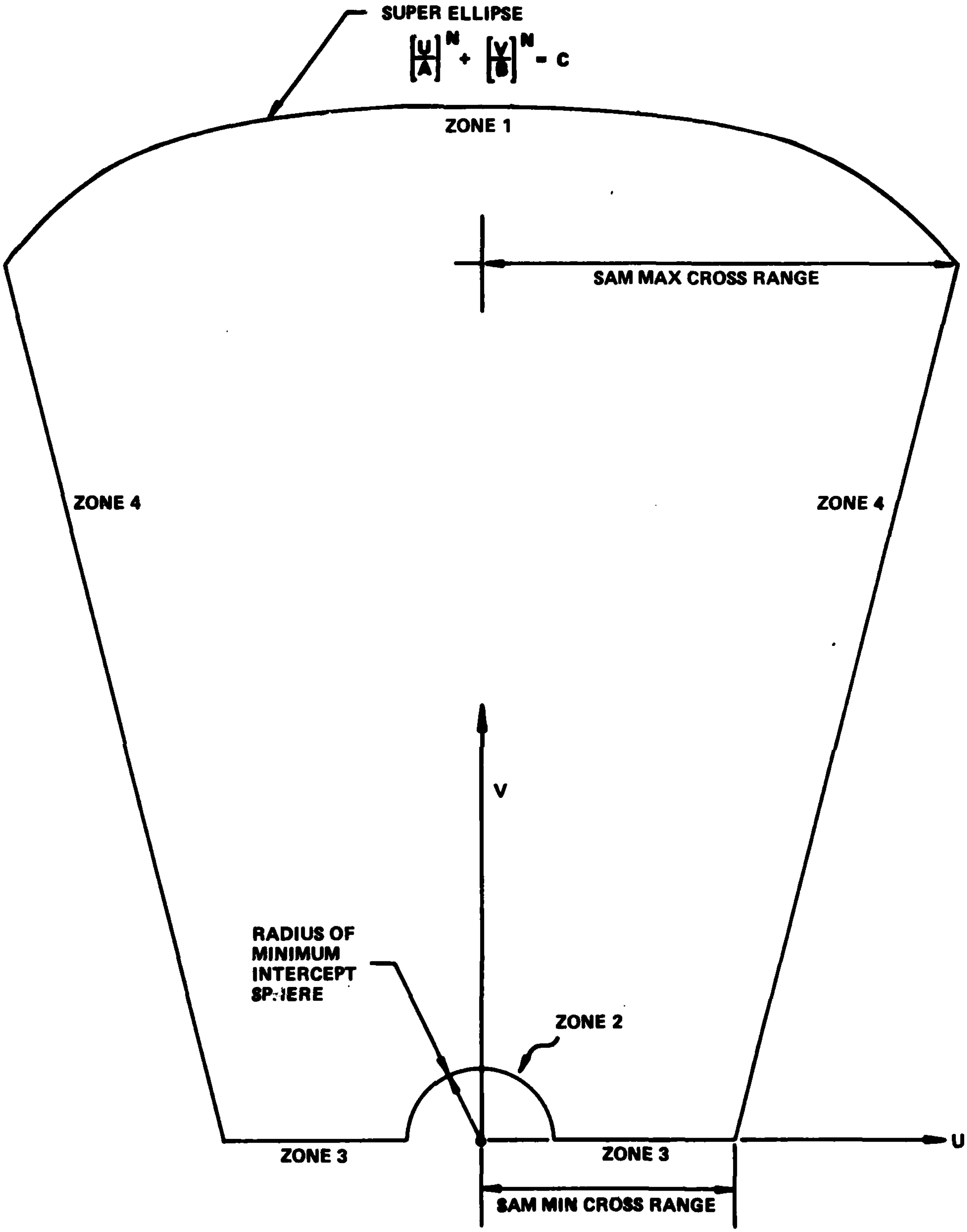


Figure 9-15. Typical SAM Envelope

second intercept. After the coordinate transformation the distances between each of these two points and the current position of the red unit are then calculated. Using these distances and the velocity of the target, the associated times to reach these points are subsequently computed and returned. The complete flow chart is shown in Figure 9-16.

9.4 LAUNCH, INTERCEPT, AND EVALUATION

The functions that control the launch, intercept, and evaluation are determined in the FORTRAN routine SAMLCH. As indicated on Figure 9-3, the logic for SAMTYP 3 differs from that for SAMTYP 1 and 2. The SAMTYP 1/2 logic is the simpler of the two and is called at the time of launch. With SAMTYP 3 the routine is called when the launcher slewing is completed. The launch may not occur at that time since it may have to wait to schedule an illuminator.

The SAMLCH routine is diagrammed in Figure 9-17. One last test is made before launch to assure that the intercept can be made before the target leaves the envelope.

At least two SAMs must be available for a salvo launch. The probability of hit (PHIT) is looked up for a single launch. For salvo launches it is recomputed. The probability of hit is tested against a uniform random number and the time of hit is set to the intercept time. If the intercept occurs in the gap, a lower PHIT is used. If the result is a hit, the threat is removed from the game at THIT, and the Fire Control Channel is released after an additional delay of TEVAL. If the outcome is a miss, nothing need happen at the intercept time. Instead the next event is at the end of the evaluation. Since the Fire Control Channel is still locked on, only a launcher is needed to fire again.

With SAMTYP 3 the subroutine SAMGT3, diagrammed in Figure 9-18, must be called to schedule the illuminator or Guidance Channel. Figure 9-2 shows that the Guidance Channel is needed time TILL before intercept. Since time has advanced to the point of being ready to launch, it is possible to compute the time it is desired to have the guidance channel available, as follows:

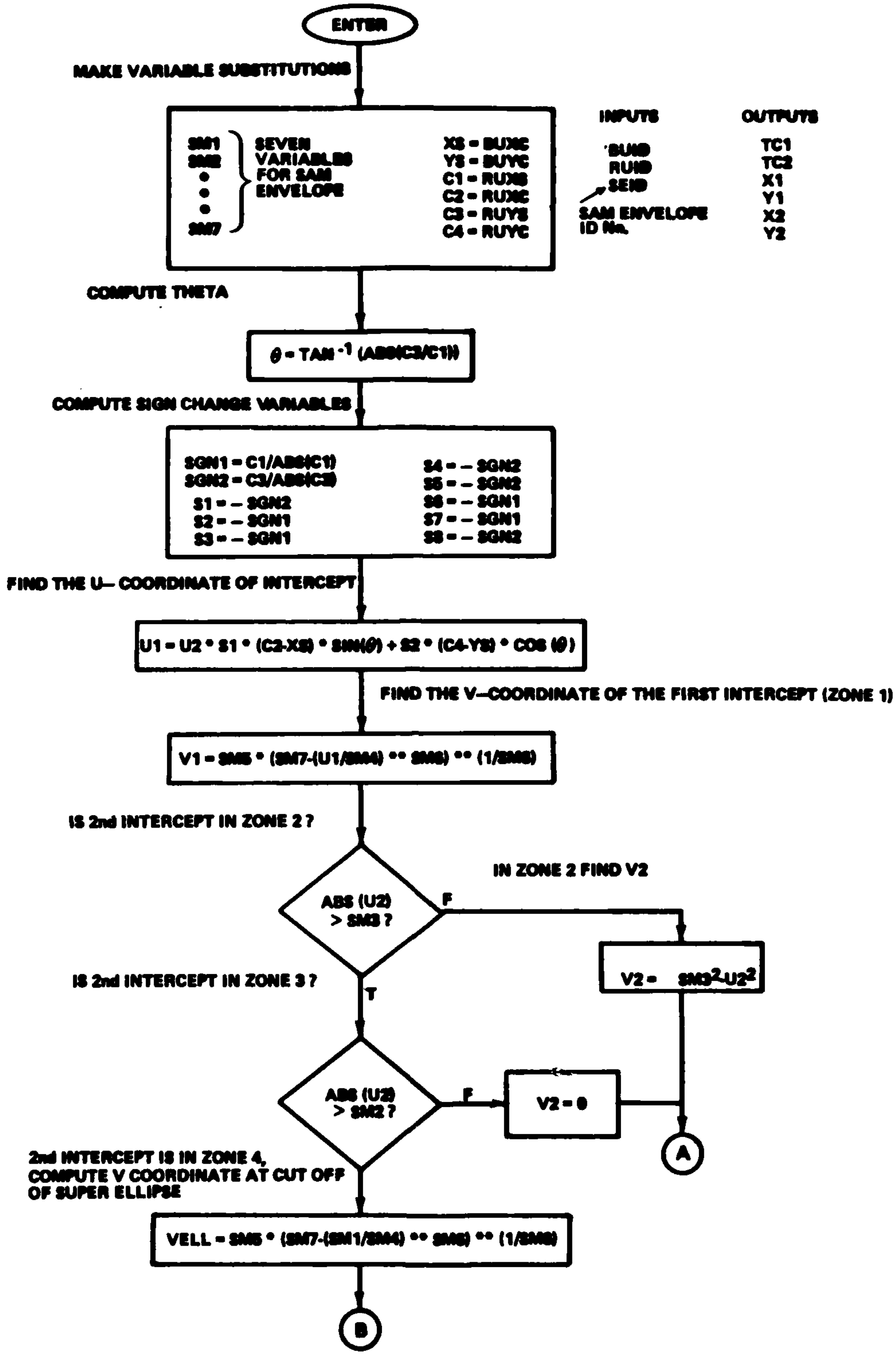


Figure 9-16. Subroutine SAMENV

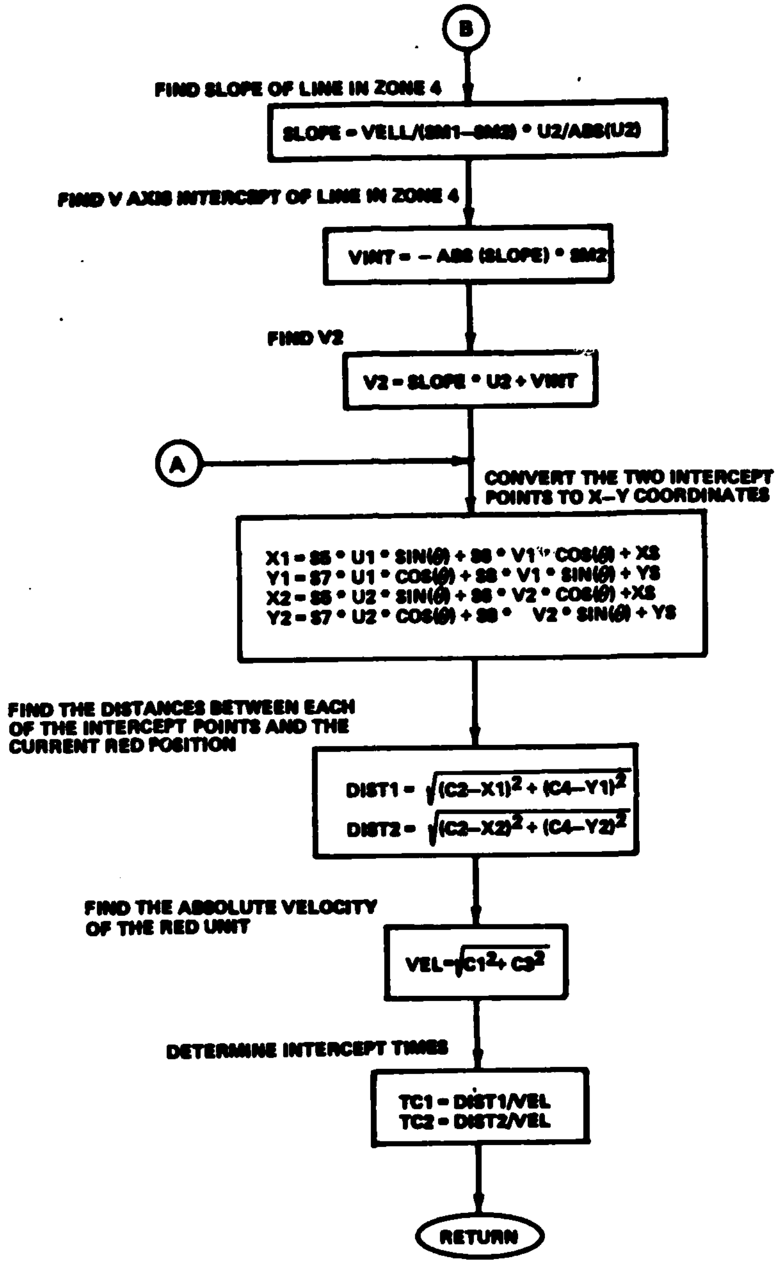


Figure 9-16. Subroutine SAMENV (Page 2 of 2)

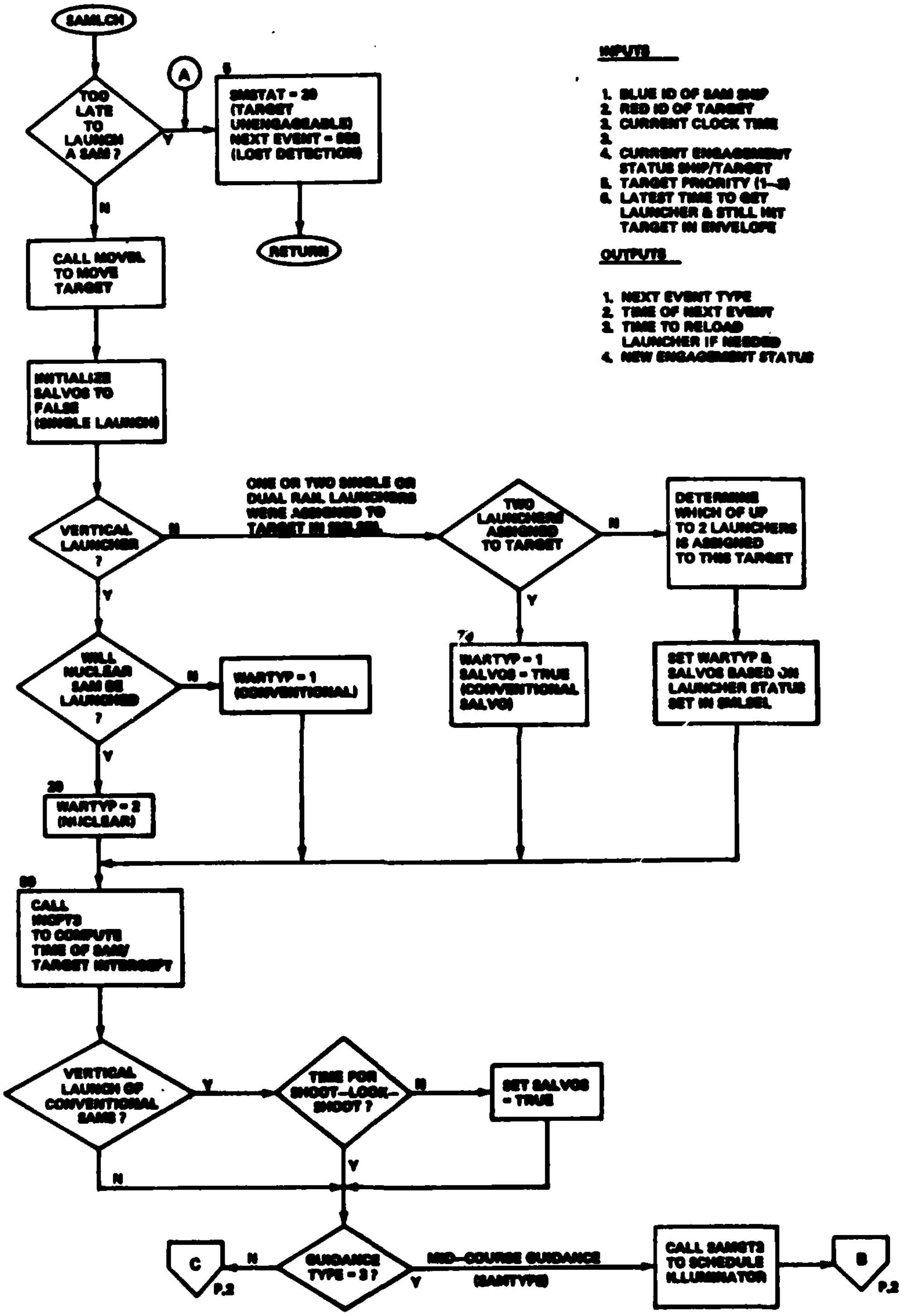


Figure 9-17. Subroutine SAMLCH (FORTRAN) (Page 1 of 2)

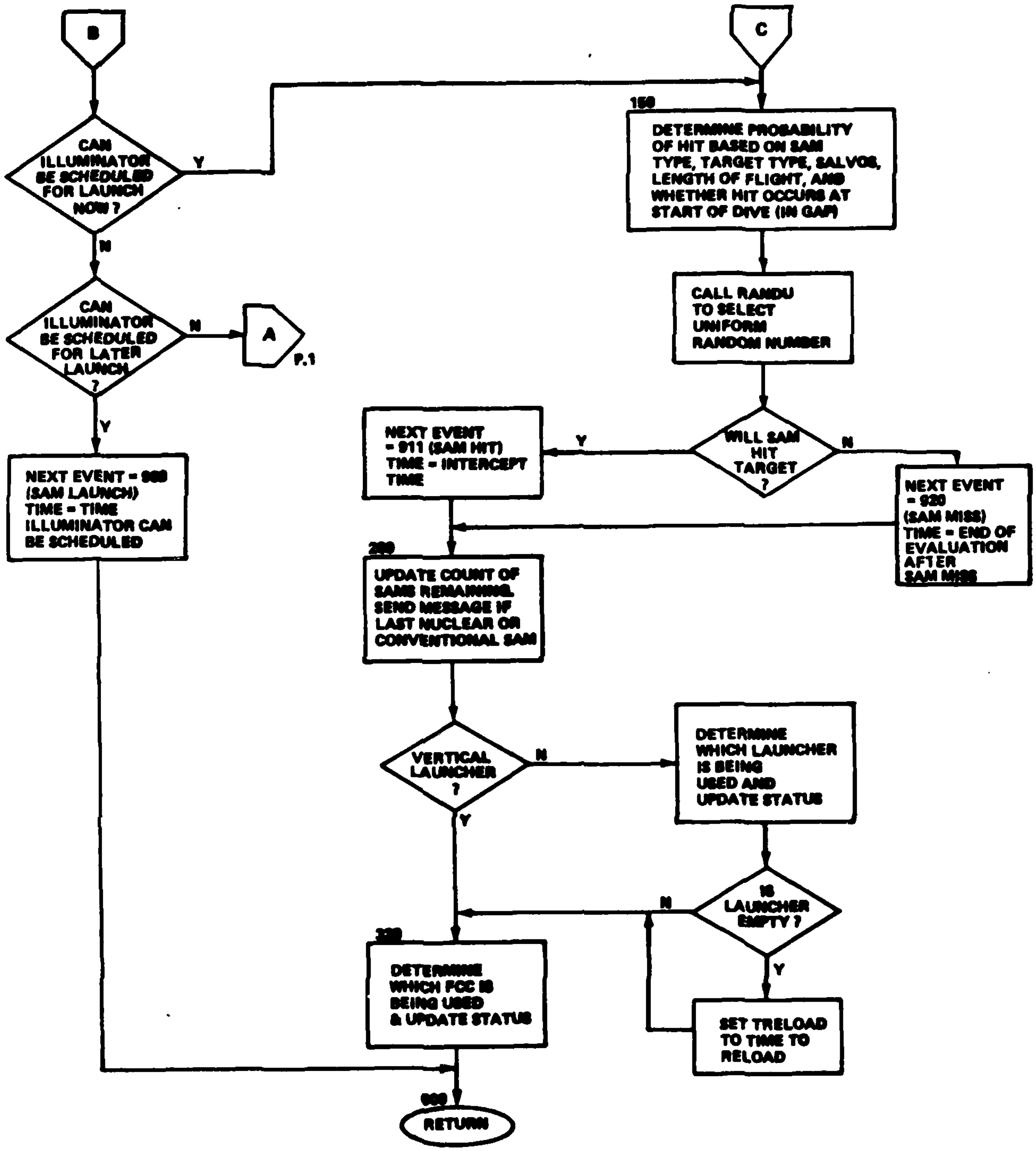


Figure 9-17. Subroutine SAMLCH (Page 2 of 2)

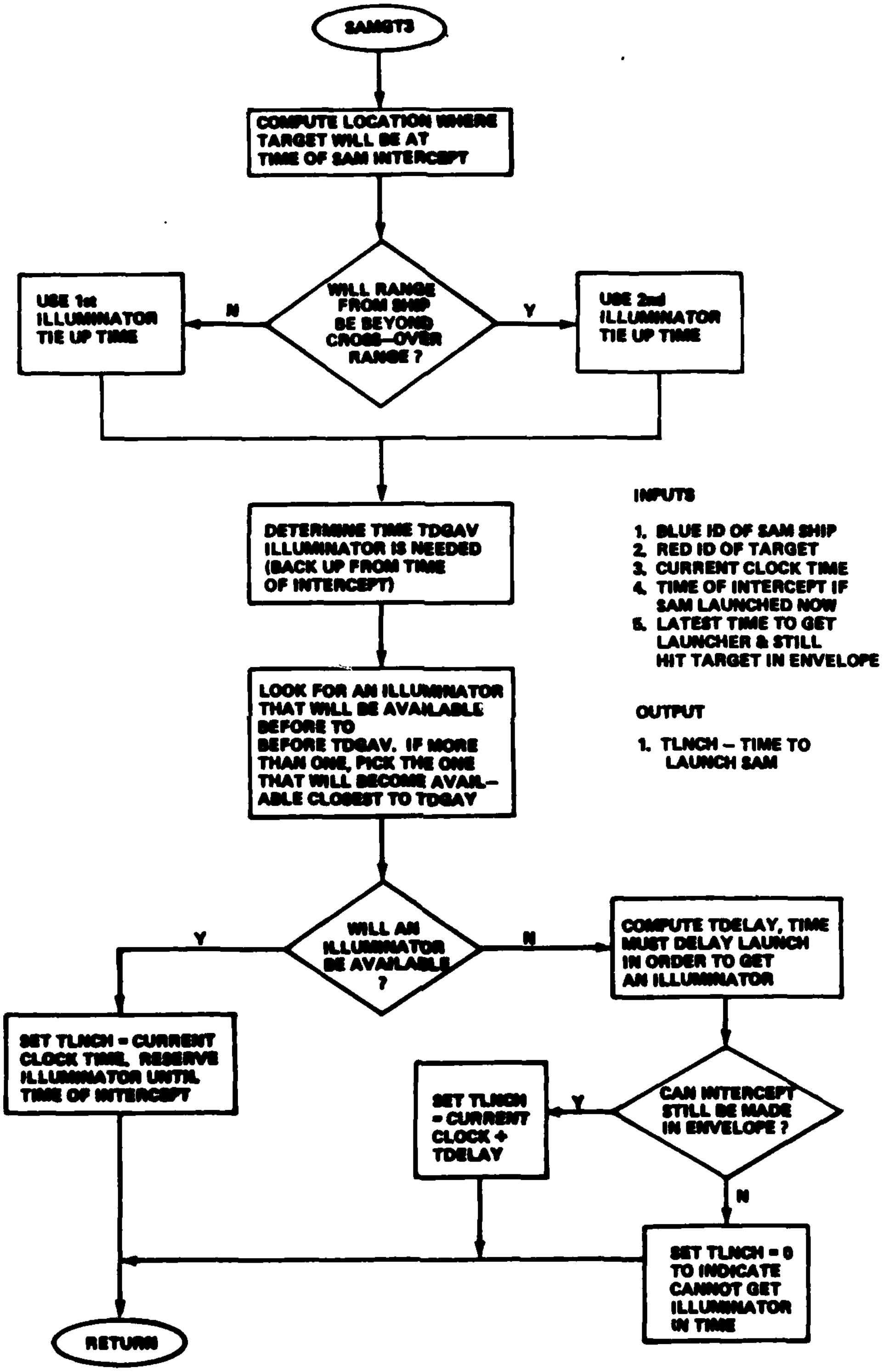


Figure 9-18. Subroutine SAMGT3 (FORTRAN)

TDGAV = TNOW + TINTC - TILL

One other system characteristic will be introduced here; at greater ranges the illuminator is on for a longer time than at short ranges. A crossover range, RILL, is input by the user as are the illuminator times themselves.

If the Guidance Channel is available at or before the desired time (TDGAV) it may be possible to launch if the intercept will occur before the target leaves the envelope. In order to determine if a long or a short illuminator time is required, the intercept range is tested against the illuminator cross-over range. If the intercept range is less, the shorter illuminator time with which the subroutine was initiated (TILL=TILL1) is used. If the intercept range is greater than RILL, then the longer illuminator time must be used and the time the illuminator is desired (TDGAV) is recalculated as well as the subsequent tests.

In situations where the Guidance Channel is not available at the earliest desired time, then the Guidance Channel availability determines when the intercept will occur. Tests similar to those just described for determining the illuminator "on" times are computed. The remaining tests and calculations determine single or salvo launches and determine hit or miss in the same manner as with SAMTYP 1 and 2.

The SAM characteristics are summarized in Figure 9-19. The first four parameters are needed for each SAM type. In addition, certain parameters are needed for each target type to be faced by the SAM. The five envelope coefficients are used to define the SAM intercept boundaries for each target type. The PHIT table defines three hit probabilities for the three corresponding SAM times of flight. If the intercept occurs between time T3 and T4 on the ASM flight, then the probability of hit becomes GAPHIT. T3 is computed when the ASM is launched, and TGAP, an input, is added to T3 to get T4.

- MAXIMUM RANGE (N.MI.)
- MINIMUM RANGE (N.MI.) (SM3)
- NOMINAL SPEED (KNOTS)
- SAMTYP – 1,2, OR 3

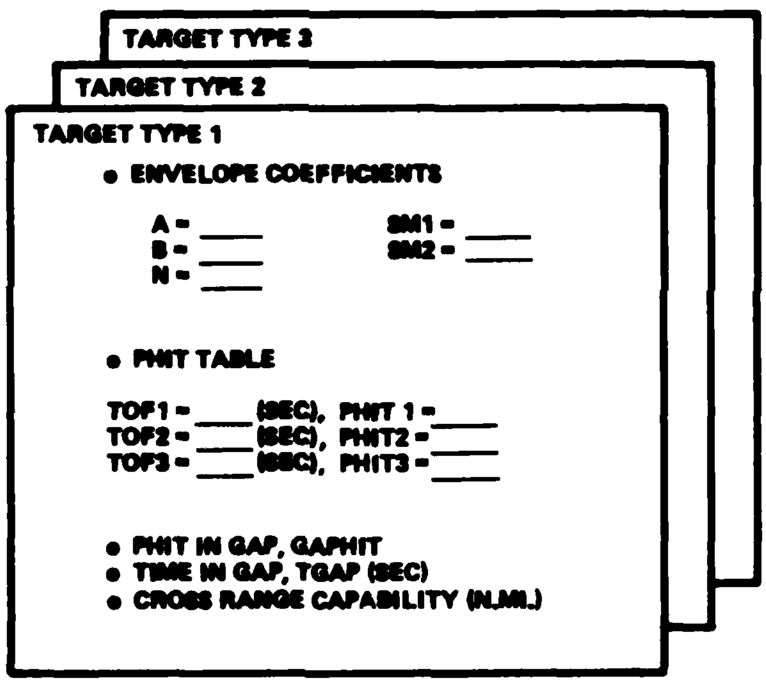


Figure 9-19. SAM Characteristics

10. CV MODULE

The Carrier Module simulates aircraft carrier flight operations necessary for Anti-Air-Warfare (AAW). It is a simple model which does not model explicitly the complex activities of the flight deck. The level of detail is mostly concerned with unscheduled operations and maintenance of an alert posture/condition CAP.

The Carrier Module generates periodic reports of the status of aircraft on each carrier, responds to orders from the command center, attempts to maintain the input alert schedule, processes returning aircraft, modifies the parameters and state of the model when damage to a carrier results, and initiates the flight of aircraft launched during the game.

Three levels of alert, together with the number of aircraft in each level, may be specified in the input to the model. Each level is the time until launch of the alert aircraft. All other aircraft in the game except those currently flying are placed in an alert category with 60 minutes to launch. An aircraft returning to the ship is placed in this category after any servicing or repair time has expired.

Eight events are unique to the CV Module: no future event; hook up; increase alert level; step down; aircraft up; land aircraft; launch aircraft; and hook up/immediate launch aircraft. The program flow for these events is diagrammed in Figure 10-1.

Hook up represents the occupation of the aircraft carrier deck and launch capability just prior to take off by an aircraft. Since each aircraft carrier is modeled as a single server facility with a mean service time, this event sequences the departure from the carriers and limits the launch rate of aircraft during the game.

The increase alert level event represents the movement of aircraft to higher alert conditions. Upon arrival at a new higher alert level the aircraft in the next alert condition are counted. If the count is less than the posture input at the start of the game, a further increase alert level event is scheduled so that the desired posture is maintained whenever feasible.

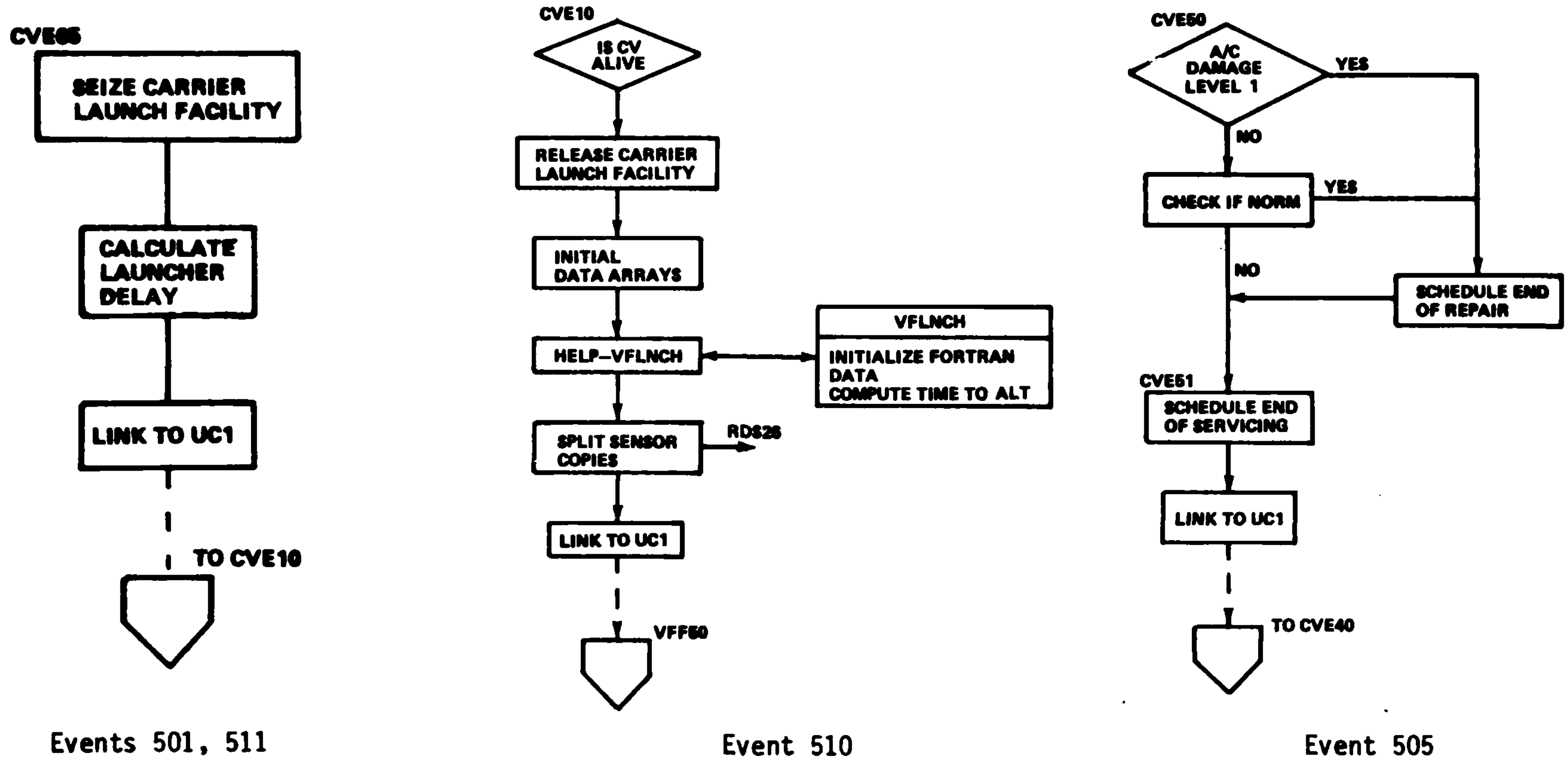
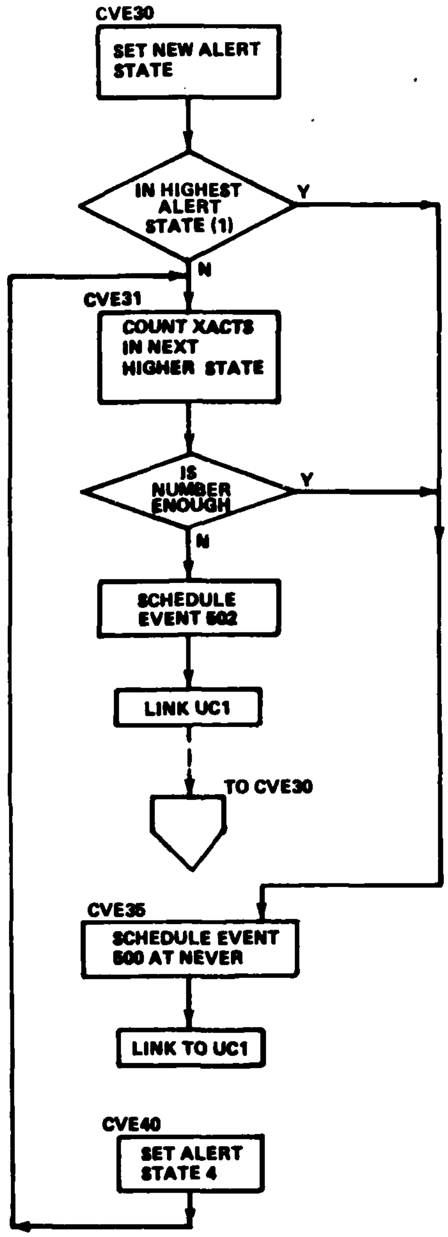


Figure 10-1. CV Module, GPSS Event Handling (Page 1 of 2)



Events 502, 503, 504

Figure 10-1. CV Module, GPSS Event Handling
(Page 2 of 2)

The aircraft up event represents the end of post flight servicing and repair. The alert posture is tested and movement through the alert states will begin if the posture is deficient.

The step down event returns the aircraft to an alert schedule upon the receipt of a cancel launch order. If preparations for launch have begun, these preparations continue until one hour after receipt of the cancelation message when the aircraft is returned to its last alert state. If launch preparation has not yet begun then the aircraft is maintained in the alert state with no action taken.

The land aircraft event represents the arrival of an aircraft on deck. This event schedules the return of the aircraft to an operational ready status. Time to repair is drawn from the exponential probability density function if the aircraft has been damaged. An independent Bernoulli trial is made to assess major failure and time to repair is drawn from the same density. Time to service is then assessed by drawing from the exponential probability density function with a different parameter. This service time should include the time for refueling, rearming, minor repair, and flight deck handling. An aircraft up event is scheduled at the expiration of the sum of the time to repair and the time to service. The launch aircraft event represents the take-off of the aircraft. This event is used to initialize the sensor copy transactions (action copy) for each red unit, to initialize appropriate GPSS arrays and to initialize data in the appropriate FORTRAN arrays. FORTRAN subroutine VFLNCH is used via subroutine HLPRTN to calculate the time the aircraft will arrive at altitude. This event is scheduled for execution in the VF module.

The CV Module processes four kinds of messages from the command center: immediate launch order; cancel launch order; revise launch order; and, deferred launch order. The program flow for these orders is diagrammed in Figures 10-2, 10-3, 10-4, and 10-5.

Upon receipt of an immediate launch order, message 110, eligible aircraft control XACTS are processed to select the aircraft with the earliest launch time. If this creates a hole in the carrier alert posture or launch plan, further processing to order alert schedule changes or scheduled launch events is performed.

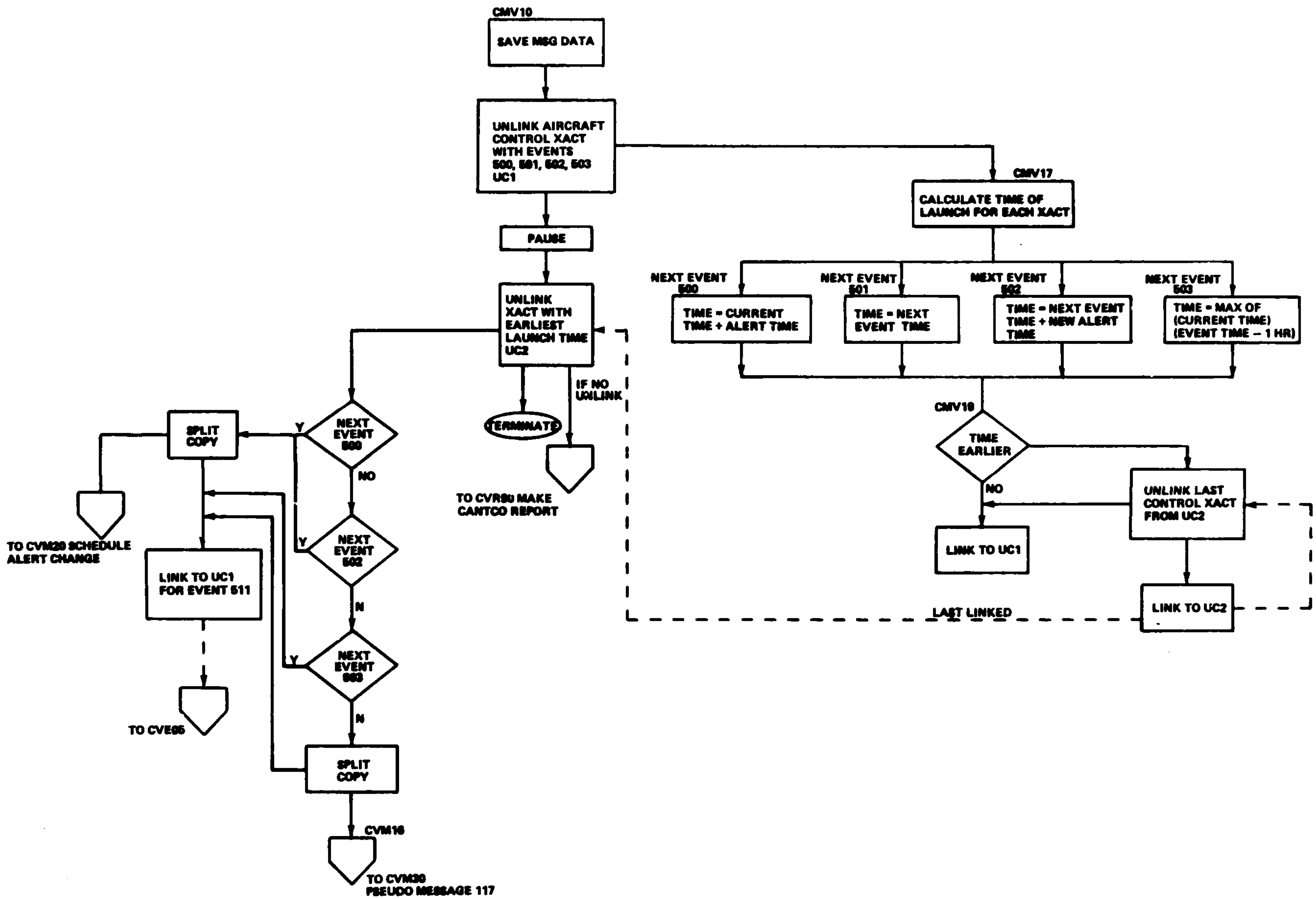


Figure 10-2. CV Handling of Immediate Launch Order, Message 110

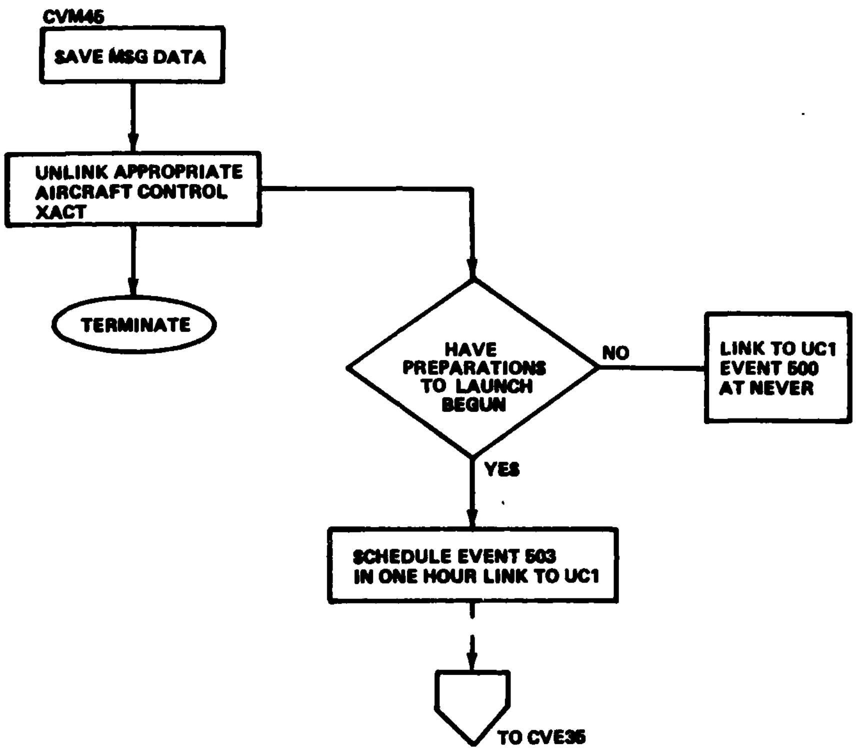


Figure 10-3. CV Handling of Cancel Launch Order, Message 114

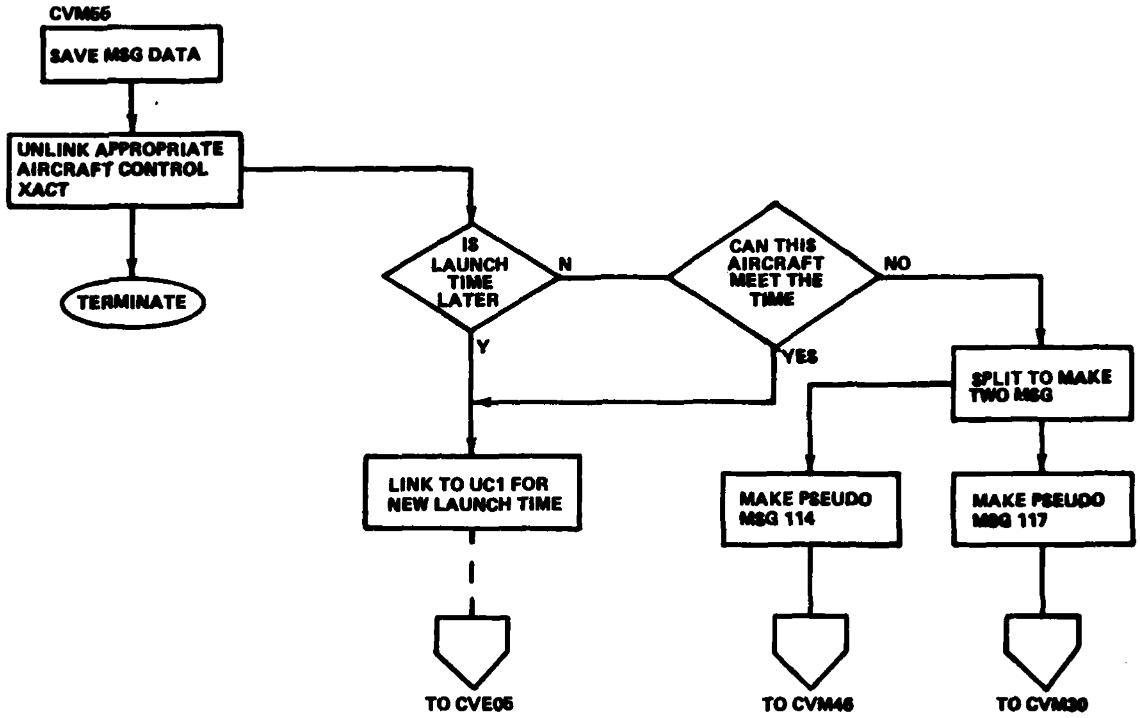


Figure 10-4. CV Handling of Revise Launch Order, Message 115

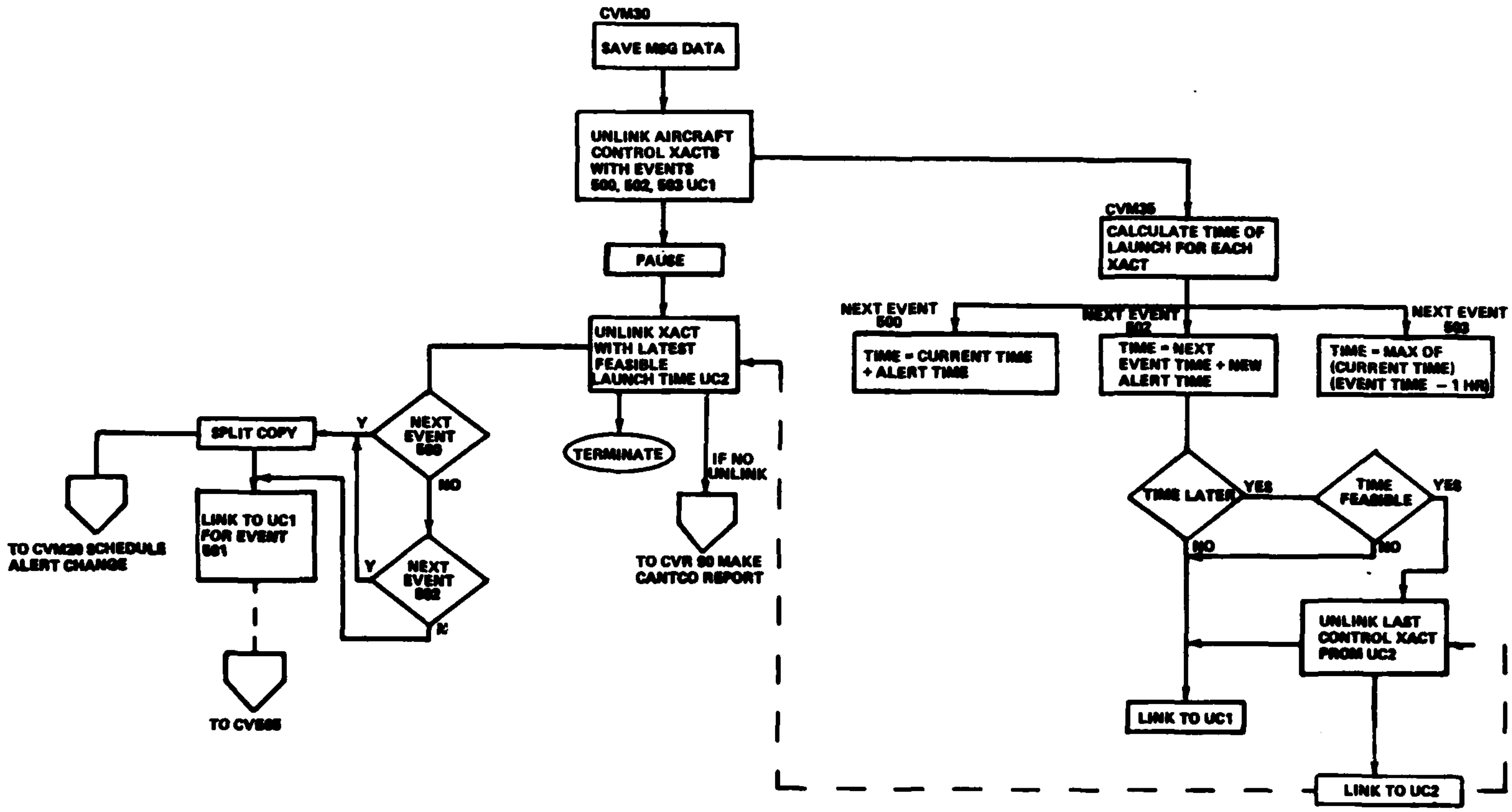


Figure 10-5. CV Handling of Scheduled Launch Order, Message 117

Receipt of a scheduled launch order, message 117, is processed in a similar fashion. The aircraft transaction selected is the one with the latest feasible launch time. Only transactions with not launch events are considered.

Receipt of a launch cancellation is processed in two ways. If the countdown to launch has begun, the countdown continues to just short of launch. The aircraft is held until an hour has expired since the receipt of the cancellation at which time the aircraft returns to the original alert state. If the countdown to launch has not begun, then no future event is scheduled. In any case, the aircraft is eligible for scheduled and immediate launch.

Receipt of a revise launch order is processed in two ways. If the launch is being delayed or if the launch is still feasible for this aircraft, a new time is assigned for the event. Otherwise, two pseudo messages are used: a cancel launch order and a schedule launch aircraft order.

The Carrier Module generates three message reports. These reports are sent to the Command Center Module and are Aircraft Status Reports, Can't Comply Reports and On-the-Way Reports.

Aircraft Status reports, message 318, are issued periodically, from all carriers about all type aircraft with a mean reporting interval of one hour. Additional reports for a specific carrier and type aircraft are issued if an order is received which the carrier module cannot satisfy.

The Can't Comply Report, message 113, is issued whenever an order or pseudo-order to the Carrier Module cannot be satisfied.

The On-the-Way Report, message 316, is issued when an aircraft is launched by a carrier.

The logical flow for each of these reports is diagrammed in Figures 10-6 and 10-7.

The Carrier Module performs special processing for damage transactions associated with a CV. With regard to flight deck operations, three damage levels have meaning: damage level one or less means no impairment; damage level two means fifty percent impairment, and, damage level three or greater means total loss of capability.

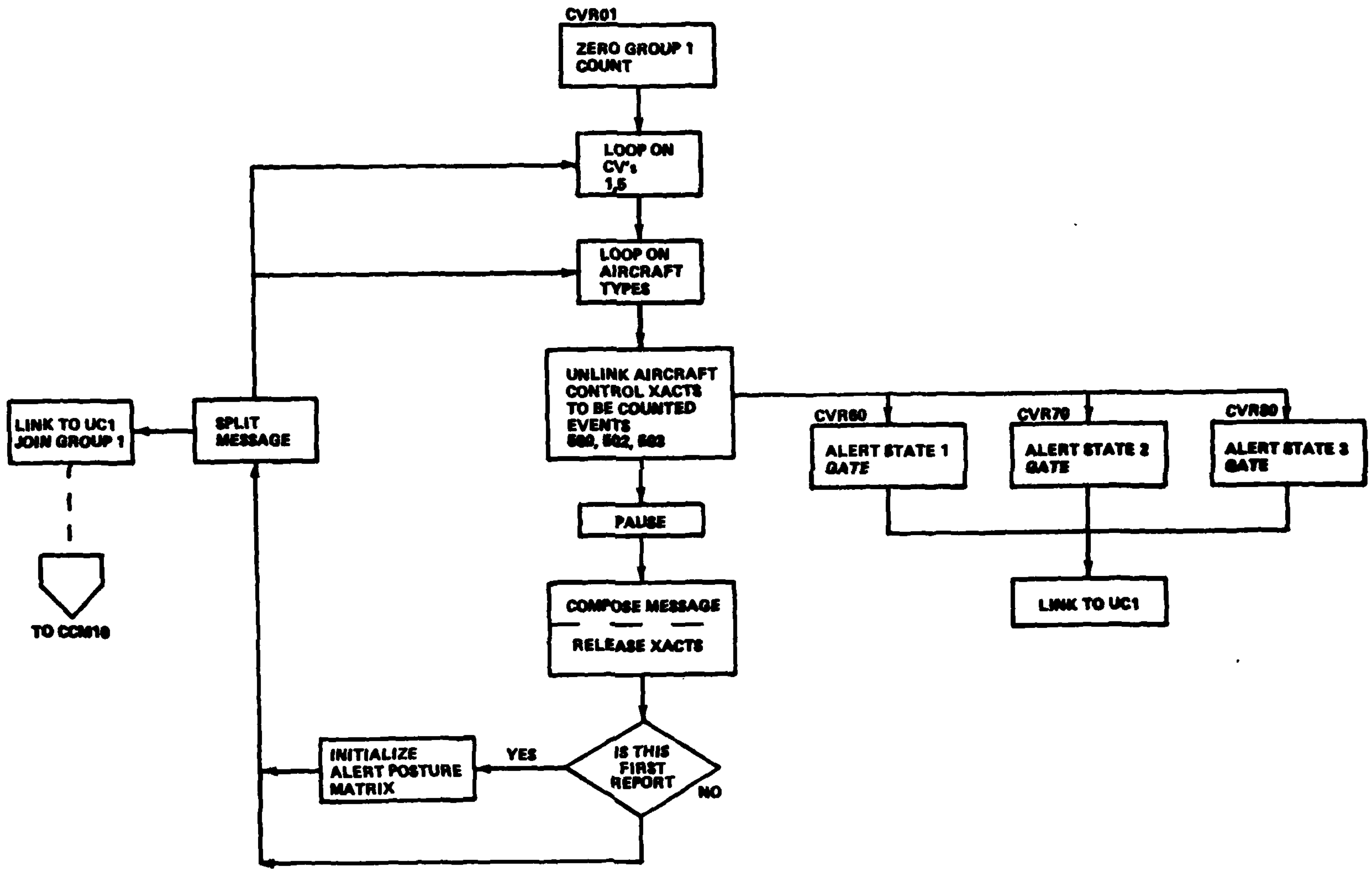


Figure 10-6. CV Generation of Periodic Status Report, Message 318

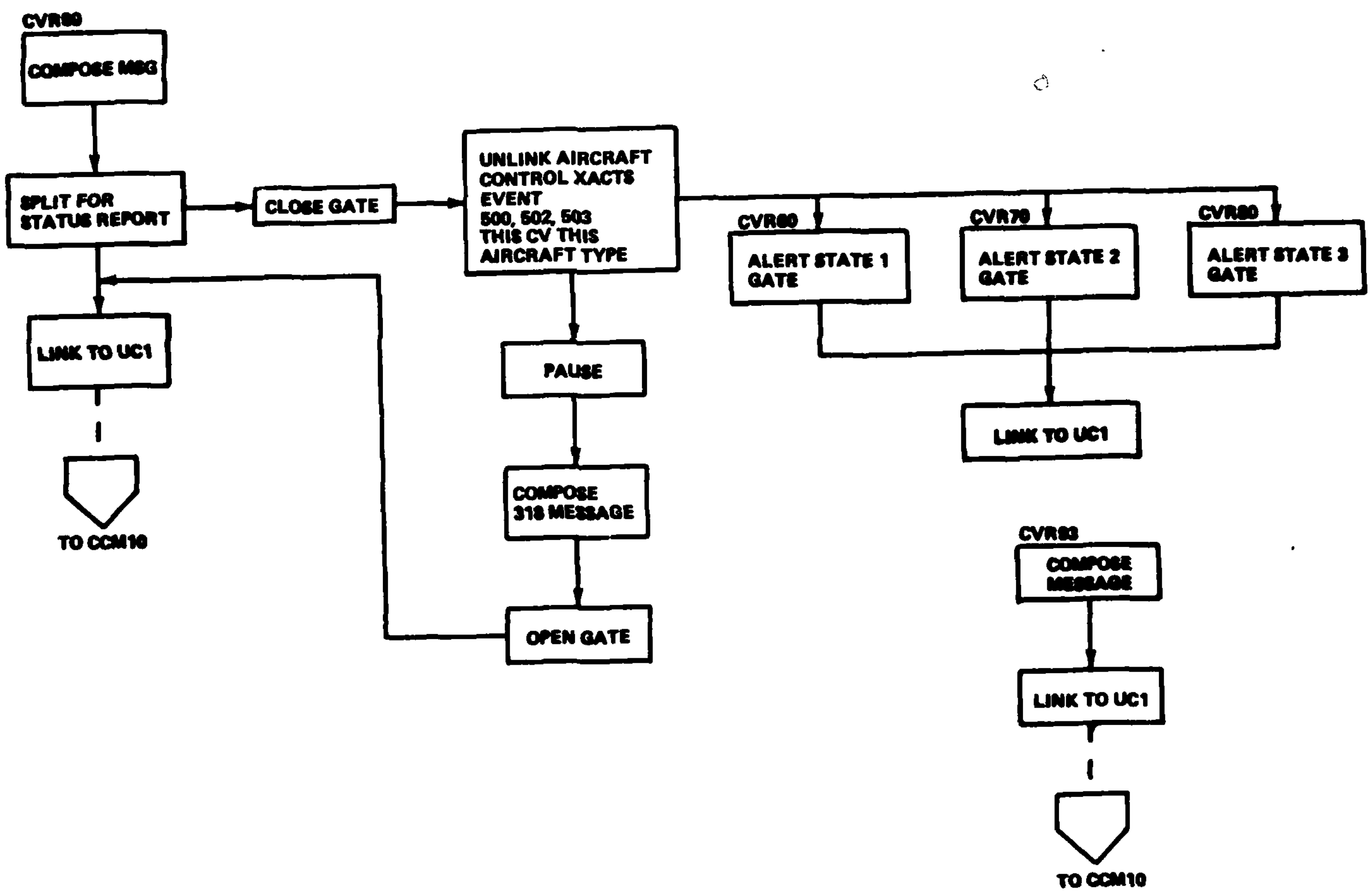


Figure 10-7. CV Generation of Messages 113, "Can't Comply", and 316, "On-the-Way"

The only complication is associated with fifty percent impairment of the carrier. Half the aircraft control transactions for aircraft on [illegible] the carrier are terminated and the interval between departures is doubled. Events are then scheduled to maintain as much of the alert schedule as is feasible with the aircraft remaining, as well as meeting the orders received from the command center.

The logical flow for carrier damage processing is diagrammed in Figure 10-8.

10.1 SUBROUTINE VFLNCH

FORTRAN subroutine VFLNCH (diagrammed in Figure 10-9) is called when a fighter aircraft is launched. It initializes new rows in the BLUNIT and VFSTAT FORTRAN COMMON arrays. It also determines the initial course for the aircraft and calculates the time, position and altitude of the aircraft at the completion of the post-take-off climb. Initial fuel states and weapon loads are input from the GPSS from parameters on the aircraft control transaction. Other characteristics are furnished from the aircraft characteristics file.

Two climb schedules are available. A normal climb is calculated if a non-zero cap station number is furnished as an input. If a Red target is furnished with no cap station number a minimum-time-to-intercept profile is used to fly the aircraft toward the target.

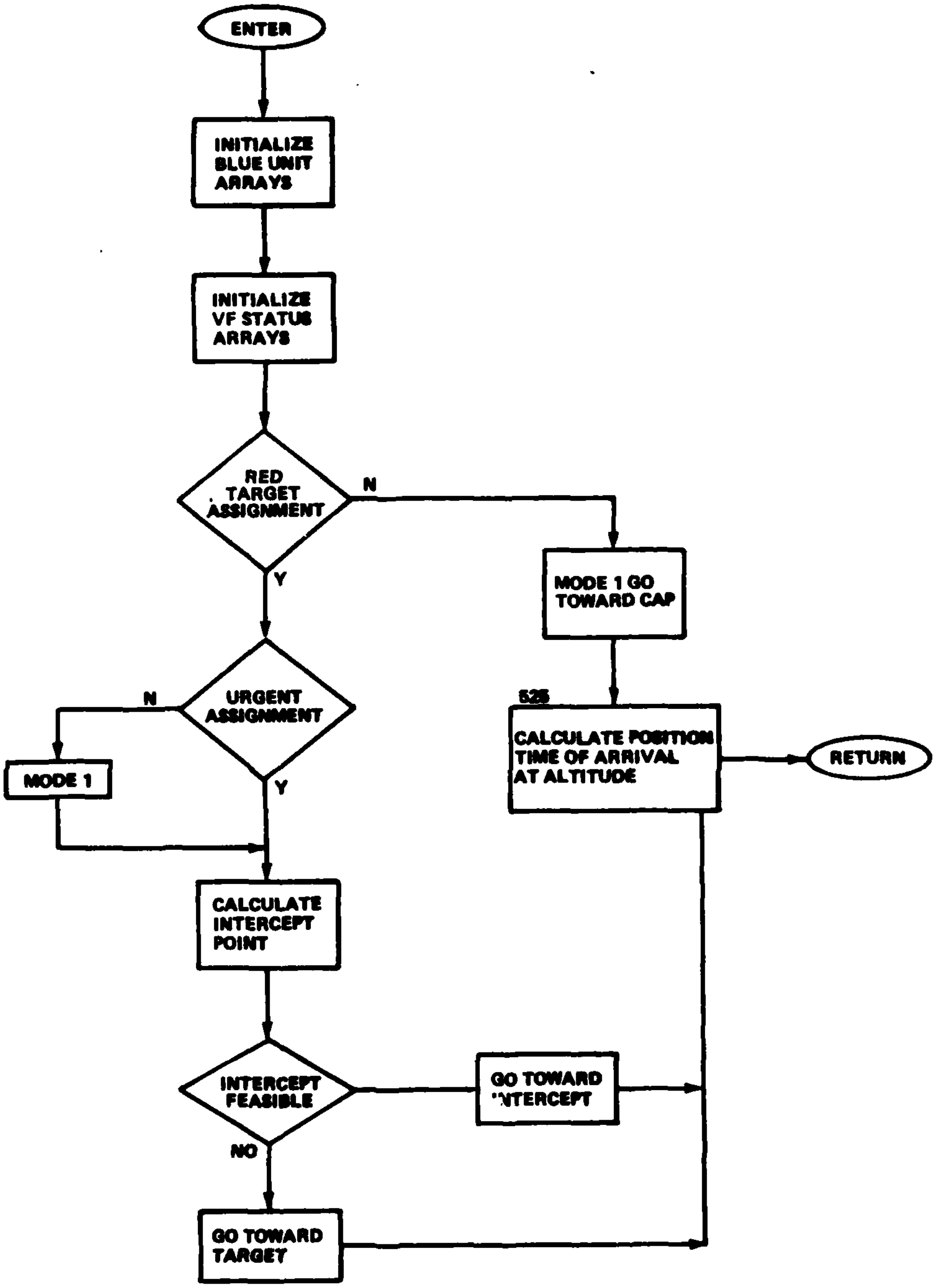


Figure 10-8. CV Battle Damage Processing

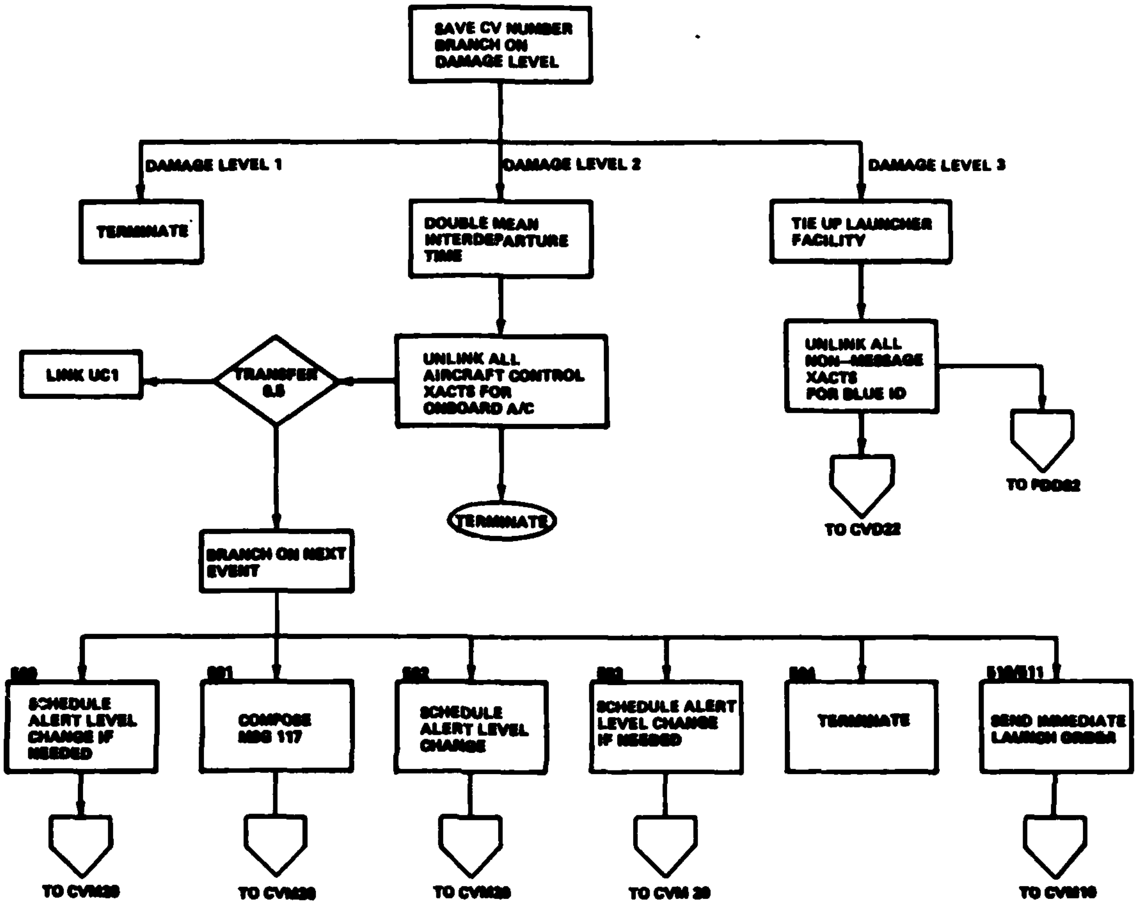


Figure 10-9. Subroutine VFLNCH

11. TERMINAL DEFENSE AND DAMAGE ASSESMENT MODULE

Battle damage is assessed during a NADs run so that the attrition of both Red and Blue capabilities can be dynamically accounted for and used to modify the ensuing course of events. Damage that VF inflict on Red aircraft and Red missiles is computed in the interceptor module. Damage that SAMs inflict on their intended targets is computed in the SAM Ship module.

The Terminal Defense and Damage Assessment module computes the damage that Red missiles inflict on the targeted ships when all defenses are penetrated, and it computes the nuclear weapon damage to all Red units and all Blue units, regardless of whether the nuclear weapon was a Red missile or a Blue SAM, and regardless of whether the Red nuclear missile was burst at its target or elsewhere as a consequence of defensive action.

The Terminal Defense and Damage Assessment module handles conventional weapons and nuclear weapons separately as described in Sections 11.2 and 11.3, respectively.

11.1 TERMINAL DEFENSE

There are two general types of Terminal Defense systems that may exist on a ship. Active systems, such as missiles and guns, produce "hard" kills. Passive systems, such as deceptive electronic countermeasure and RF decoys, produce "soft" kills. Existing and planned active systems are: Basic Point Defense Missile System (Sparrow), NATO Sea Sparrow Surface Missile System, Close-In Weapon System, and the 5-inch dual mode missile. The passive systems are not discussed here because of their security classification. Any given ship may use one or more active or passive systems which makes it difficult to model the entire Point Defense system as an aggregate system.

Some of the factors that affect the performance of each system are as follows:

- o Attack density
- o Reaction time
- o Kill distance from ship
- o Fratricide
- o Jamming
- o Missile or ordnance supply
- o Weather
- o Search radar and sensor integrations

All of these factors are wrapped up in a probability of kill (PKPD) figure for the combined Point Defense system against specific ASM types and a random time delay. Figure 11-1 illustrates the general implementation of the Point Defense model as well as the Damage Assessment, which is described later.

The number of ASMs penetrating to each ship is scored by ASM type. ASM type is available as RUMTY in the RDUNIT common. ASM type is used to identify nuclear ASMs and to determine Point Defense outcome. ASMs that are defeated are tested to see if they have conventional warheads. Nuclear armed ASMs are passed to damage assessment even if the Point Defense is successful.

Damage by conventional ASMs is evaluated as described in Section 11.2. The nuclear ASM damage is evaluated to a greater level of detail and is done by NUCLER, as described in Section 11.3.

11.2 CONVENTIONAL WEAPONS

Each non-nuclear ASM that reaches a ship will be evaluated to determine what damage it will cause to the ship. Only those systems simulated in NADS are examined. The design is intended to be simple and yet accommodate multiple ship configurations. The functions that may be damaged are: aircraft launching (CVs only), detection capability, SAM launching capability, and the Point Defense System.

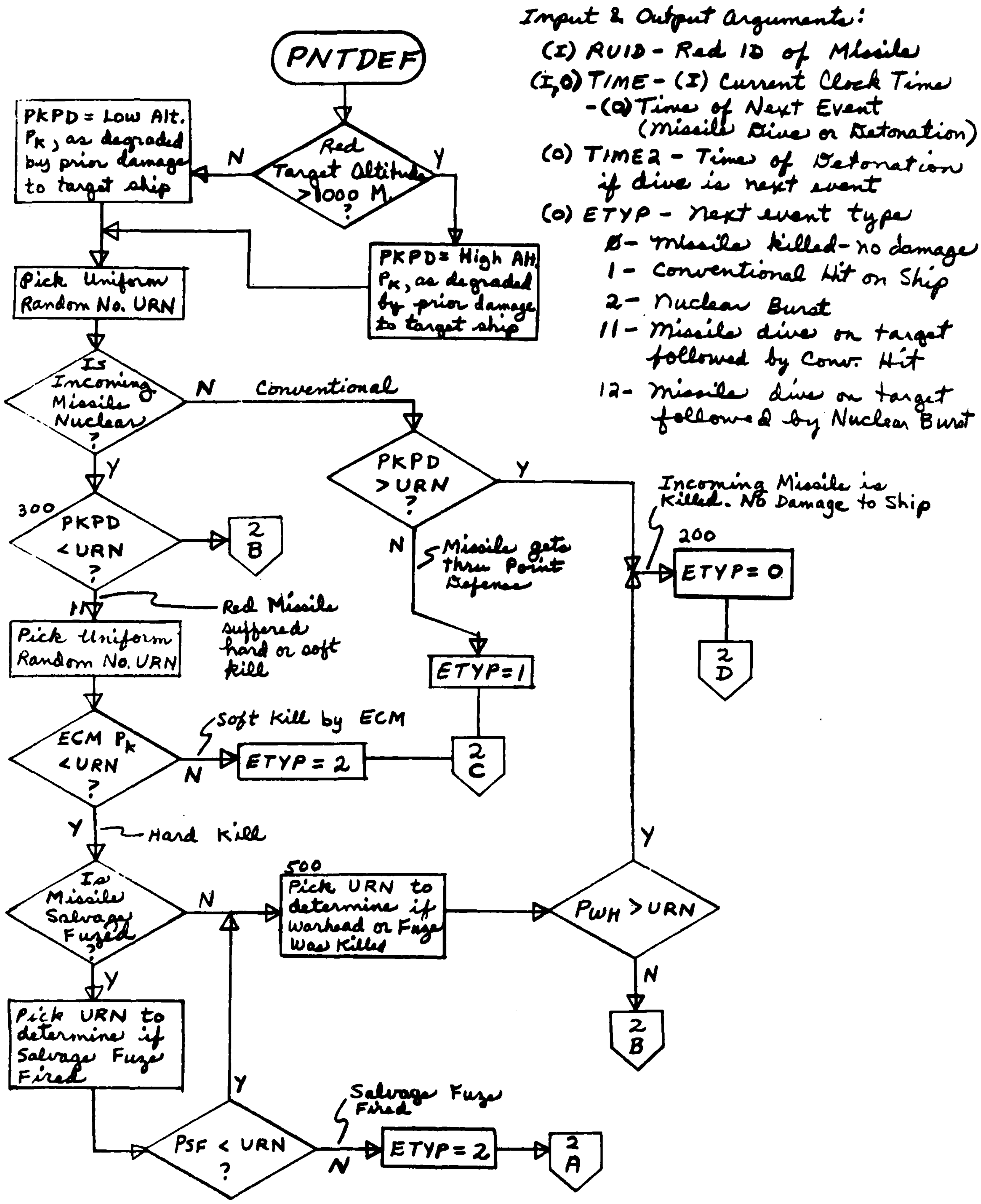


Figure 11-1. Point Defense and Damage Assessment

(Page 1 of 2)

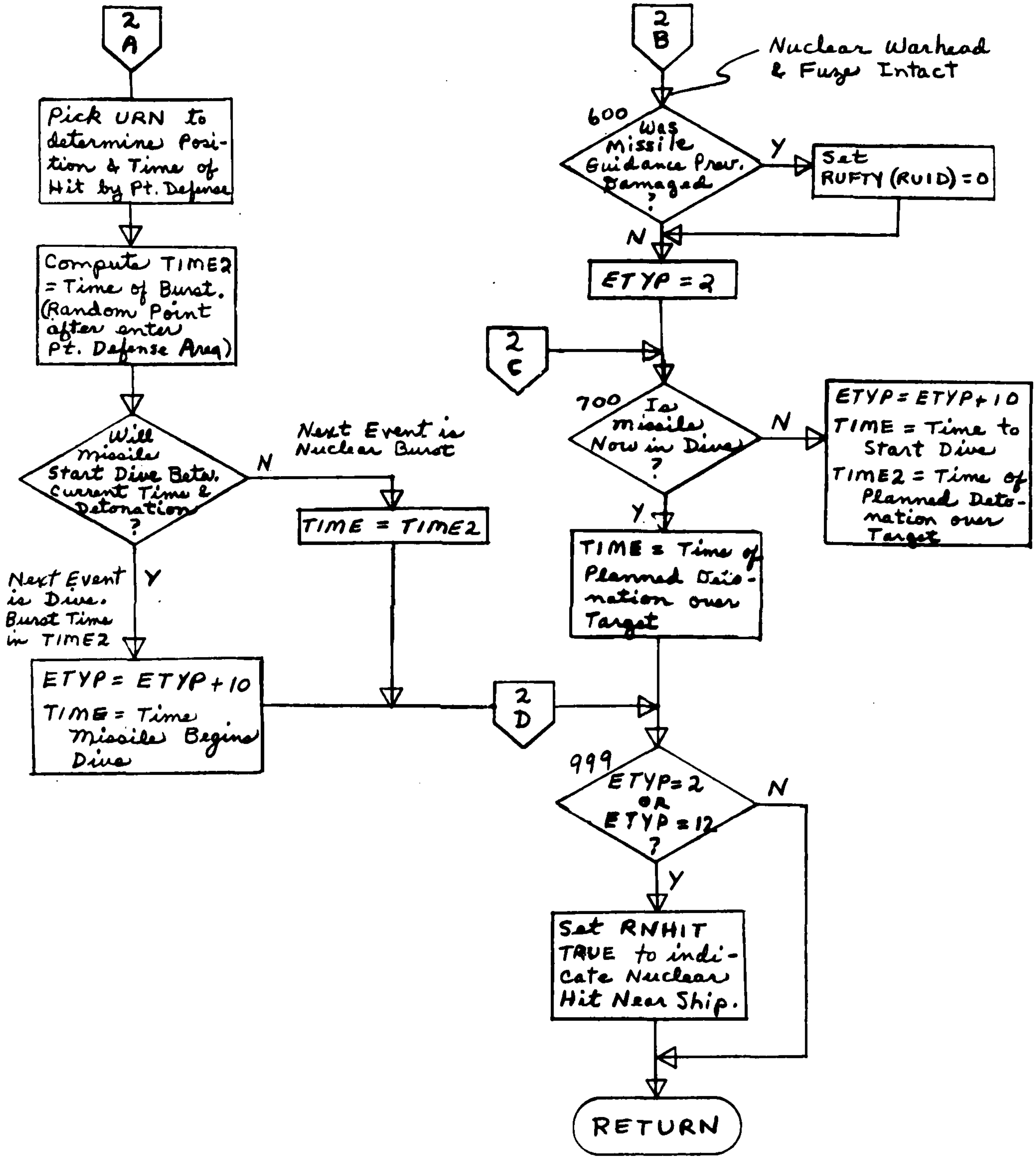


Figure 11-1 Continued. Point Defense and Damage Assessment

(Page 2 of 2)

The general scheme is to have three possible operating levels each for aircraft launching and SAM launching, and two possible operating levels each for detection capability and Point Defense System. The CV may have normal aircraft launching capability with four catapults available, reduced capability with two catapults available, or none available. These availabilities establish the Alert-1 capacities in the CV Operations module. The SAM launching capability is similarly conceived except the number of launchers may differ from ship to ship. The launcher acts as a proxy for the entire SAM system. A fully "up" system on ships that have either two launchers or a vertical launching system (VLS) is the normal initial state. The reduced capability leaves the ship with one launcher or one-half of the VLS cells. A ship with a single launcher starts in this state. The third level is no SAM launching capability. Both the detection and Point Defense capabilities are either normal (fully "up") or none.

A system is changed from one level to another after the number of penetrators exceeds the value input by the user specifying how many hits are required to change levels. Penetrators are simply counted until they exceed the input threshold, then the system capability is reduced. Because the SAM system is so dependent on the detection system, loss of the detection system automatically reduces the SAM capability by one level.

11.3 NUCLEAR WEAPONS

Whenever a Red or Blue nuclear warhead burst occurs, subroutine NUCLER is called to ascertain the consequent damage to all Red and Blue units. Because NUCLER is a lengthy procedure (it calls 20 other subroutine and function subprograms), a significant portion of it is devoted to editing out the cases that needn't be computed.

NUCLER uses a number of FORTRAN common blocks to obtain most of its input data and to store all of its output data. Its specific calling arguments are a Blue ID number, a Red ID number, a flag indicating whether the Red ID is the bursting warhead or the target, and the time of the burst.

It is assumed that there are four basic circumstances in which NUCLER will be called:

CASE 1: Red nuclear missile penetrated all defenses and burst at its intended X-Y coordinates and design HOB. The targeted ship is presumed sunk without further computation.

CASE 2: Red missile suffered previous airframe and/or guidance system damage, but its warhead and fuzing system is unimpaired. It bursts at the design HOB but at an erroneous X-Y position.

CASE 3: Red missile was salvage fuzed, which could occur at any point on its flight path if it was defined as having salvage fuzing.

CASE 4: Nuclear SAM was burst at the position of its intended target. The SAM ship module will already have determined the success or failure of its shot, and it is assumed here that it was successful or the warhead would not have been given a detonation command. Such an assumption is required because NADS does not maintain position data along SAM trajectories.

In all four cases, it is the X-Y-Z position of the Red ID that defines the position of the burst. The Blue ID is needed only to look up the warhead characteristics for a Case 4. (The Blue ID is the launching ship, which yields successively the ship class, the SAM type number, and the warhead type number, IWHB.)

In the initial implementation of NADs there are eight nuclear environments (weapons effects) that are individually evaluated and compared with relevant threshold values for damage assessment. The eight environments, which are identified in the flow charts and the code by the subscript J are:

1 Blast Overpressure
2 Blast Dynamic Pressure
3 Overpressure Impulse
4 Particle Velocity, Normal to flight path
5 Particle Velocity, Parallel to flight path
6 Prompt Gamma Peak Dose Rate

7 Neutron Fluence

8 Thermal Fluence

The damage threshold values against which the computed environment levels are compared are stored in the several FORTRAN common blocks that represent the characteristics of the various types of missiles and platforms in the game. The general form of the response (or vulnerability) arrays is

XXRSP(N,J,K)

where the XX indicates the particular block of characteristics data, e.g., RMRSP for data on Red Missiles. The N is the index to a specific platform type, the J is each of the nuclear environments, and K is an arbitrary designator for the level of damage. The range of K varies among the different functional types. For ships, K goes up to 6 representing successively increasing levels of damage, viz:

K=1 Loss of air search radar

K=2 50% weapon delivery impairment

K=3 100% weapon delivery impairment

K=4 90% mobility impairment

K=5 90% seaworthiness impairment

K=6 Ship destroyed (sunk)

As an example if SHRSP(4,1,2) = 8.5 it indicates that a ship whose class is indicated by 4, if exposed to blast overpressure (1), will suffer a 50% weapon delivery impairment (2) if the overpressure exceeds 8.5 psi. (It also loses its radar.)

Red missiles are characterized by only three levels of damage: 1 is airframe or guidance damage that would neutralize a nonnuclear missile but not a Nuc, 2 fires the salvage fuzing (if any), and 3 disables the warhead. Aircraft have only two levels of damage: 1 is loss of mission capability, and 2 is loss of the aircraft.

NUCLER's use of subroutines, function subprograms, and labelled common blocks is illustrated in Figures 11-2 and 11-3. These are followed by the flowcharts that were used to define the code. Several of the subroutines have been extracted from earlier studies by TRW and others, and their flow charts were not available.

The general scheme of NUCLER (Figure 11-4) is to first establish the burst position and the type of warhead and log the burst for post-simulation review. The effective blast yield is then computed and a maximum range of potential damage is estimated as a basis for excluding unnecessary computations. Beginning on sheet 2 of the flowchart, every Red unit and every Blue unit is considered as a potential victim of each of the nuclear environments. The full list of potential victims is divided up into three major groups because of the different array names and array dimensions that must be referenced. The first group is all Red units, covered by sheets 2 and 3. The second group is all Blue units except VF on sheet 4 and the top of sheet 5. The third group is the VF covered in the remainder of the 7 sheets.

The same pattern of computations is used in each of the three groups. A succession of tests is applied to screen out as many units as feasible from the more detailed computations. Each unit is eliminated from further consideration by the following tests:

1. It has already left the game.
2. It is more than four times the range limit and thus could not intercept the shock within the range limit.
3. It is beyond the range limit and its velocity is not bringing it closer fast enough to meet the shock within the range limit.
4. If it is beyond the range limit at burst time it will not be subject to radiation effects, regardless of its velocity.

If a potential victim cannot be elimінted by the simple tests, then the appropriate subroutines are called to compute the intensity of each of the environments for which the platform type has a vulnerability criterion specified in the XXRSP tables. The computed intensity for each unit is then compared with the criteria for each damage level, and the maximum damage level for each unit is recorded along with the particular environment (J) that inflicted the maximum damage and the time it happened.

When NUCLER has done that for all the environments against all the victims, it returns to the calling program.

CALLED \ CALLER		SUBROUTINES						FUNCTIONS						
		BLAST	BLINT	FENV	NUCLER	PULSE	SCALE	AGC	PMS	RHOX	RSHK	RTPP	RTRIR	TSHK
SUBROUTINES	BLAST				X									
	BLINT				X									
	FENV				X									
	MATM62						X			X	X			X
	PULSE	X												
	SCALE	X												
	VPARTS	X												
FUNCTIONS	AGC				X									
	PFA	X							X					
	PMS	X												
	POLYNF							X						
	RHOX				X									
	RSHK		X											
	RTPP	X												
	RTRIR	X												
	TERPL			X						X				
	TERPL2					X								
	TERPT								X			X	X	
	TSHK				X									

Figure 11-2. Subroutines and Function Subprograms Required to Support NUCLER

COMMON BLOCK LABELS

	RDUNIT	RACHAR	RMCHAR	BLUNIT	VFSTAT	BACHAR	SHCHAR	SMCHAR	NWCHAR	ENV	BLCOM	SCALT	NUCLOG	NWO
NUCLER	X	X	X	X	X	X	X	X	X	X			X	
AGC														
BLAST		X	X			X	X			X		X		
BLINT										X				
FENV									X	X				X
MATM62														
PFA														
PMS														
POLYNF														
PULSE										X				
RHOX														
RSHK														
RTPP														
RTRIR											X			
SCALE												X		
TERPL														
TERPL2														
TERPT											X			
TSHK														
VPARTS										X				

Figure 11-3. Common Blocks used by NUCLER and its Subprograms

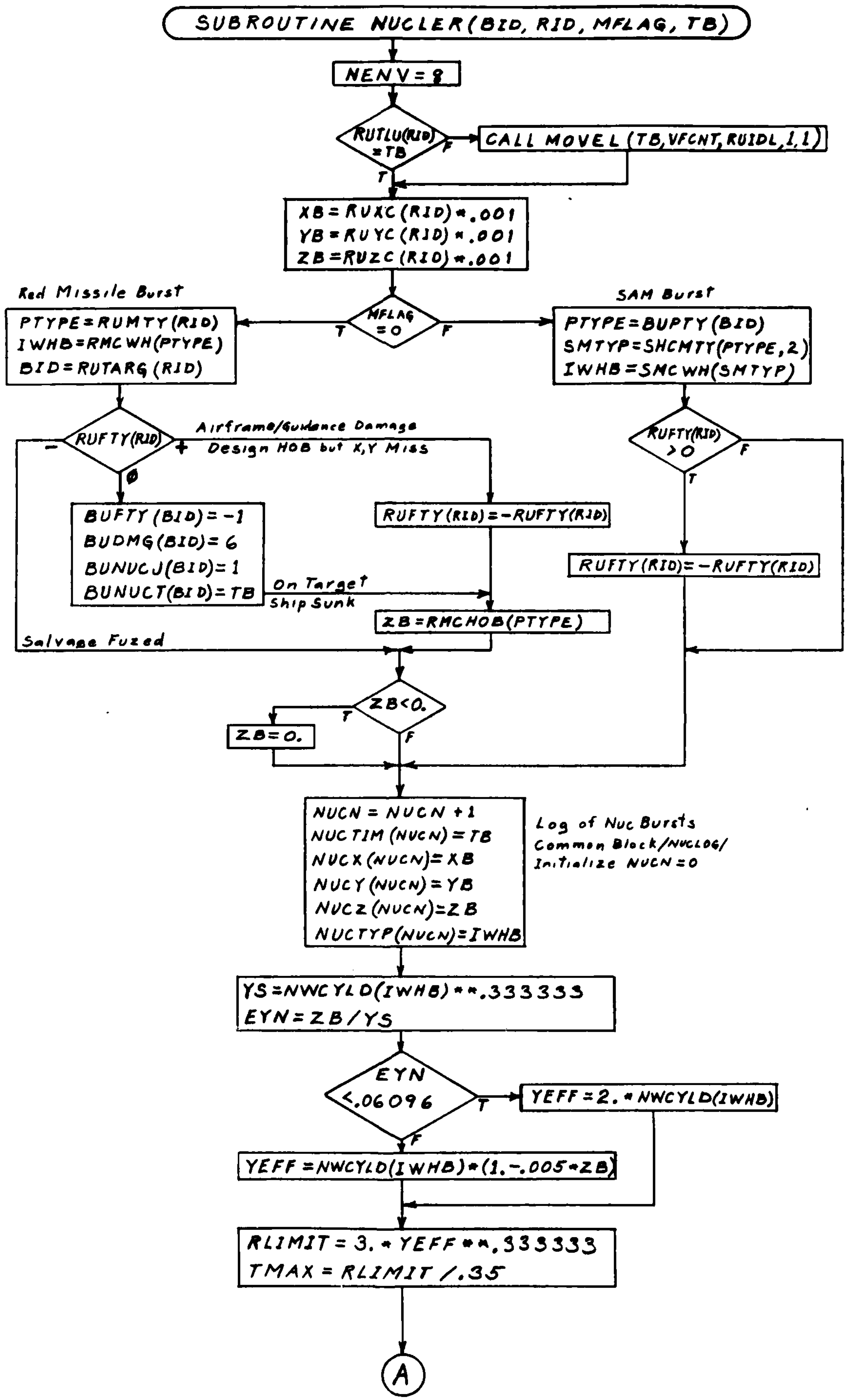


Figure 11-4. Subroutine NUCLER (Page 1 of 7)

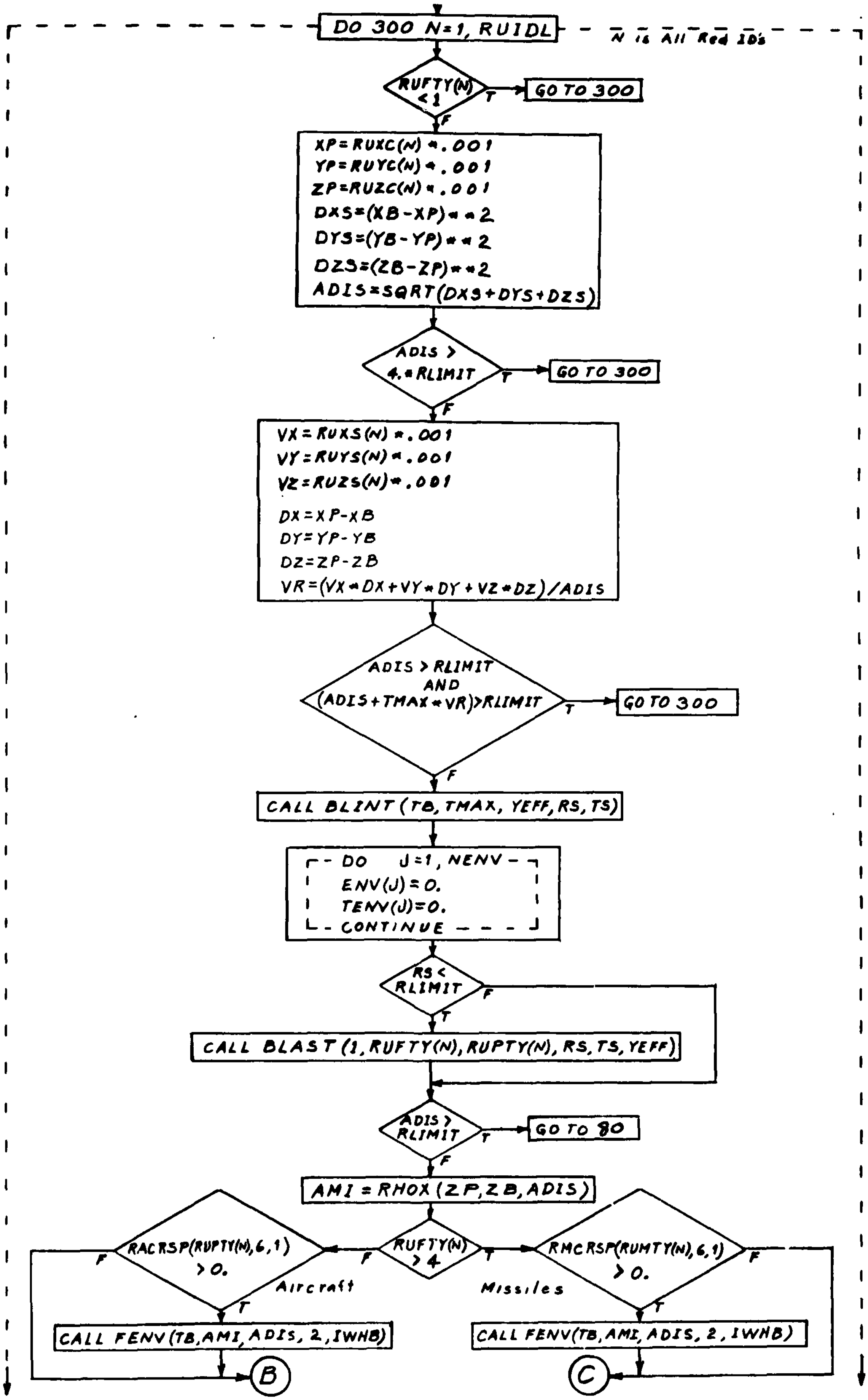


Figure 11-4 Continued. Subroutine NUCLER (Page 2 of 7)

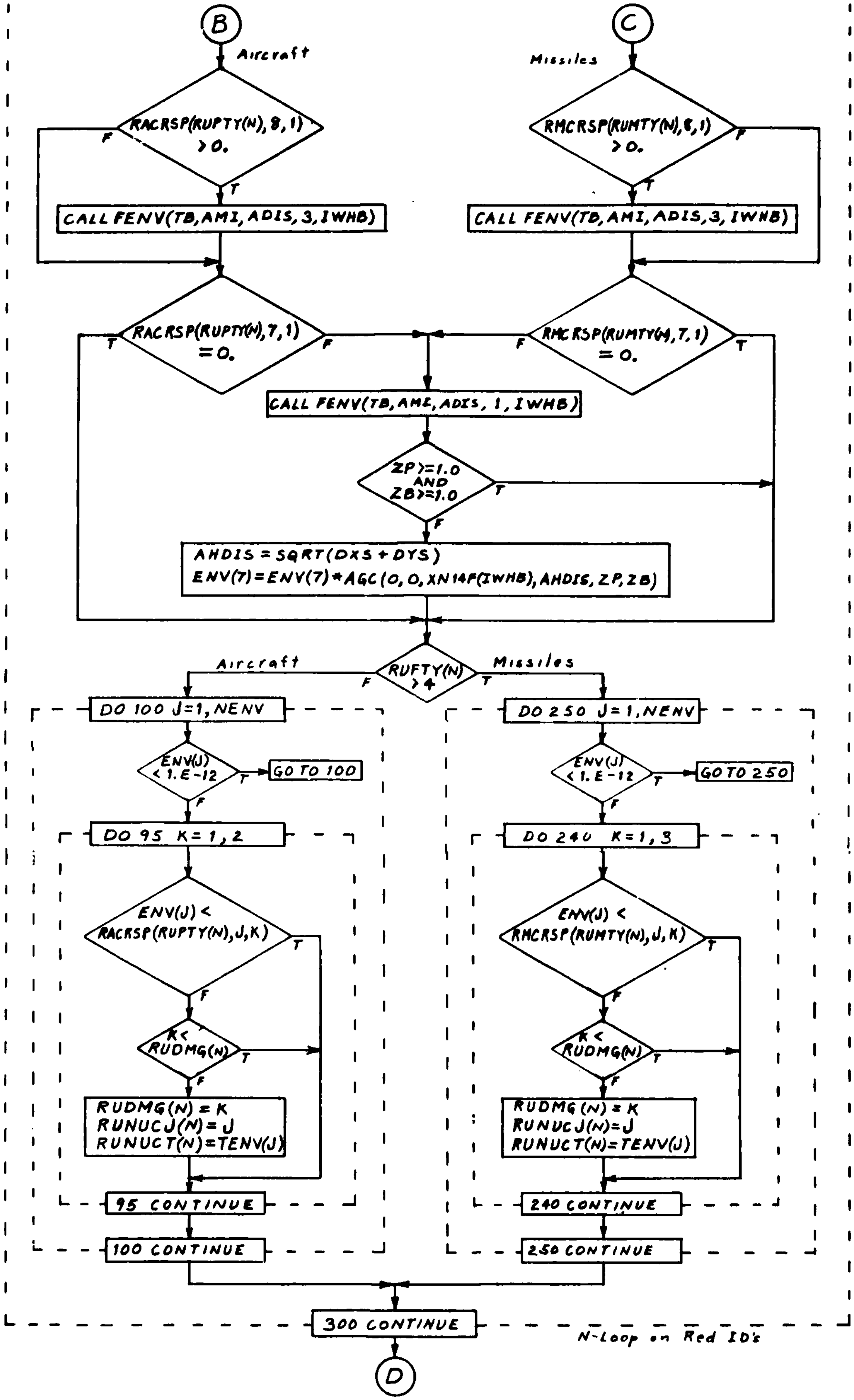


Figure 11-4 Continued. Subroutine NUCLER (Page 3 of 7)

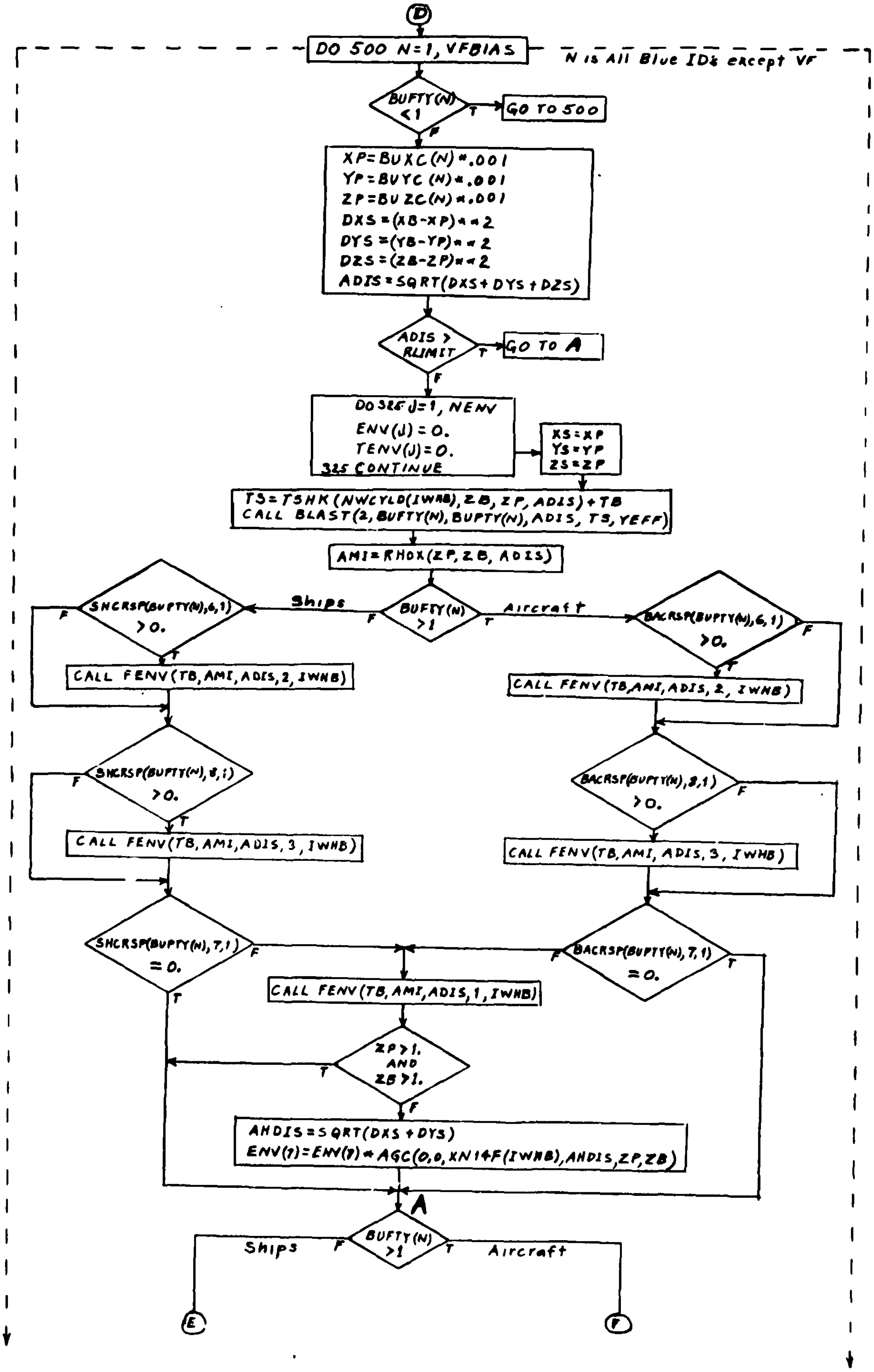


Figure 11-4 Continued. Subroutine NUCLER (Page 4 of 7)

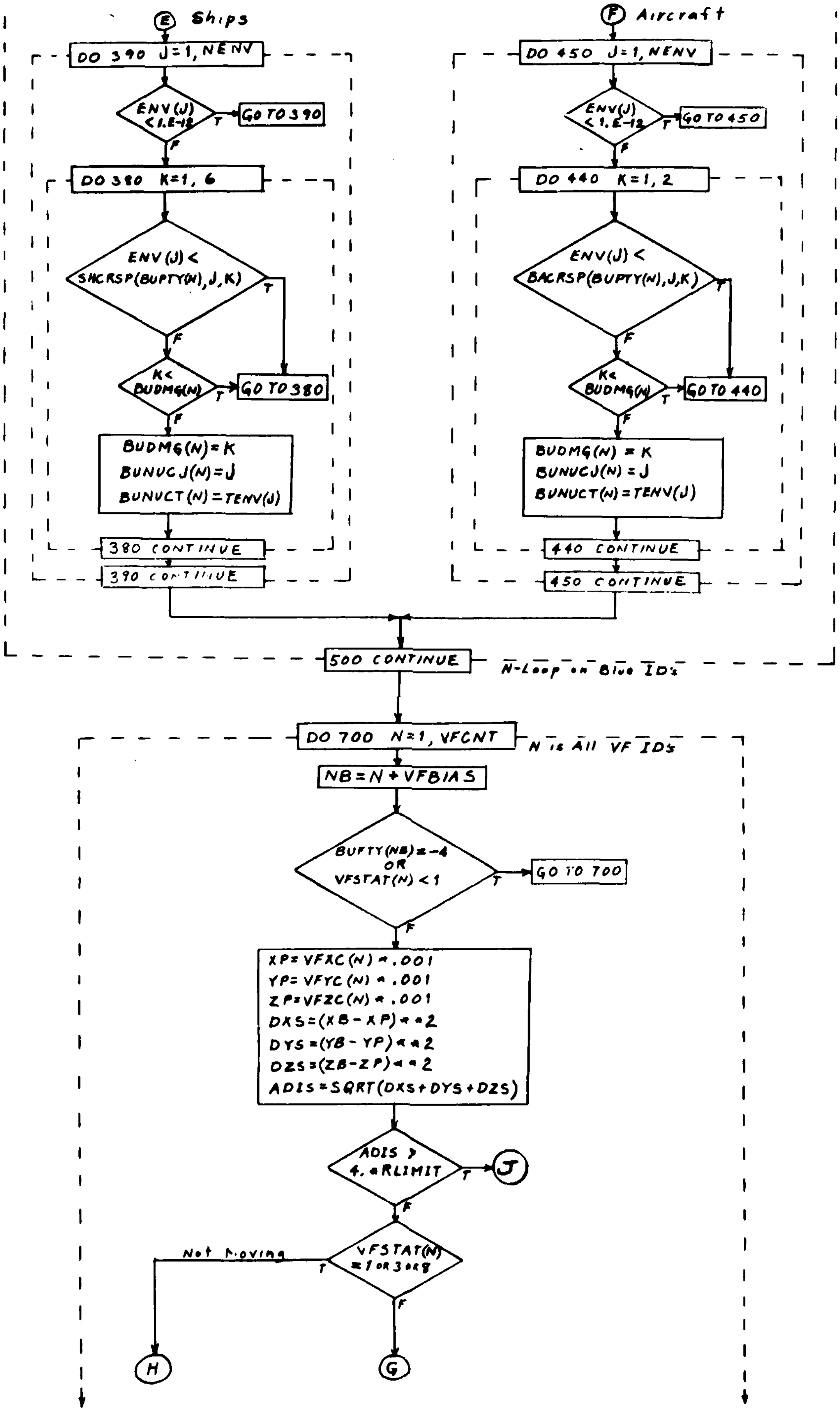


Figure 11-4 Continued. Subroutine NUCLER (Page 5 of 7)

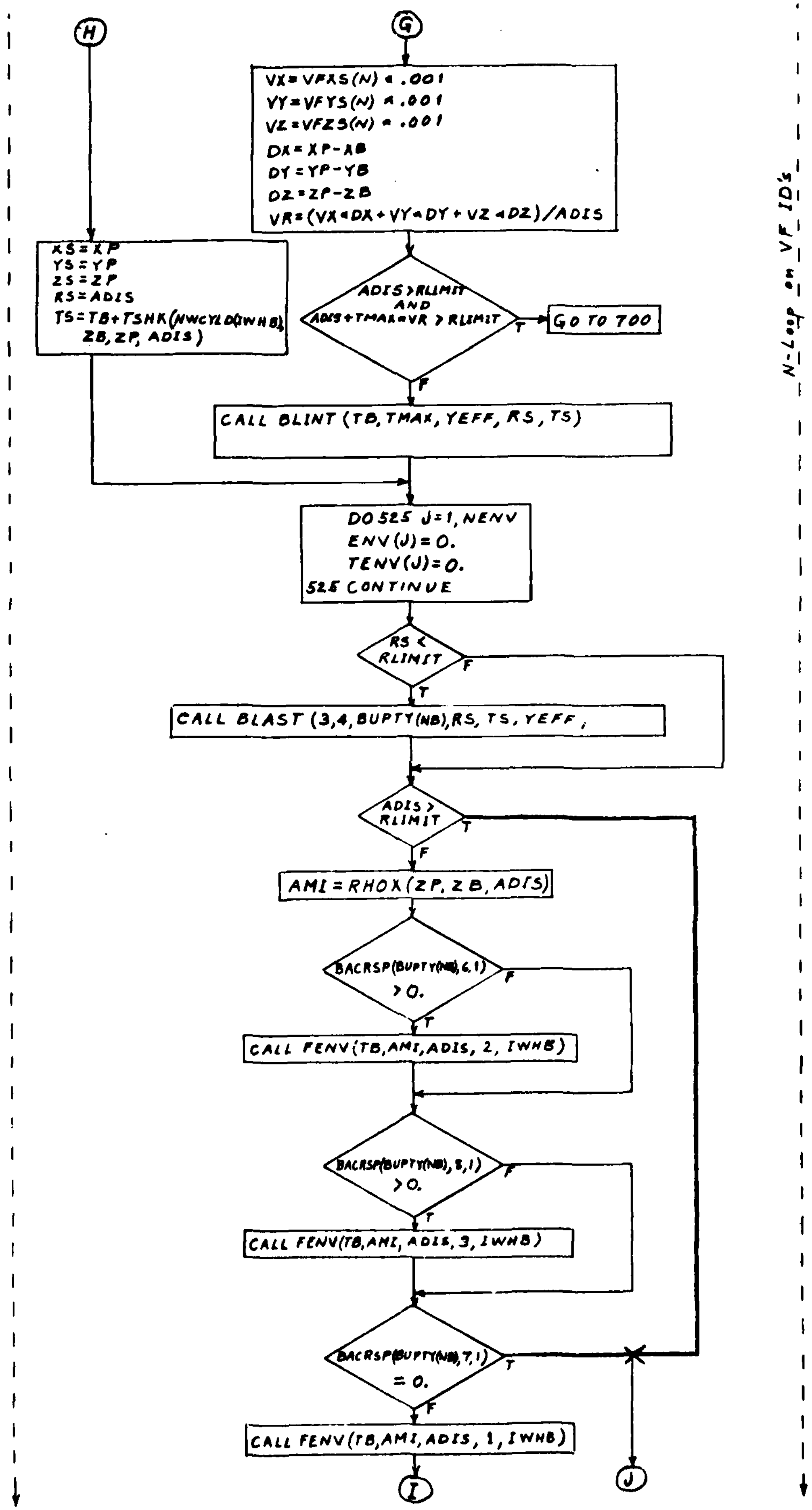


Figure 11-4 Continued. Subroutine NUCLER (Page 6 of 7)

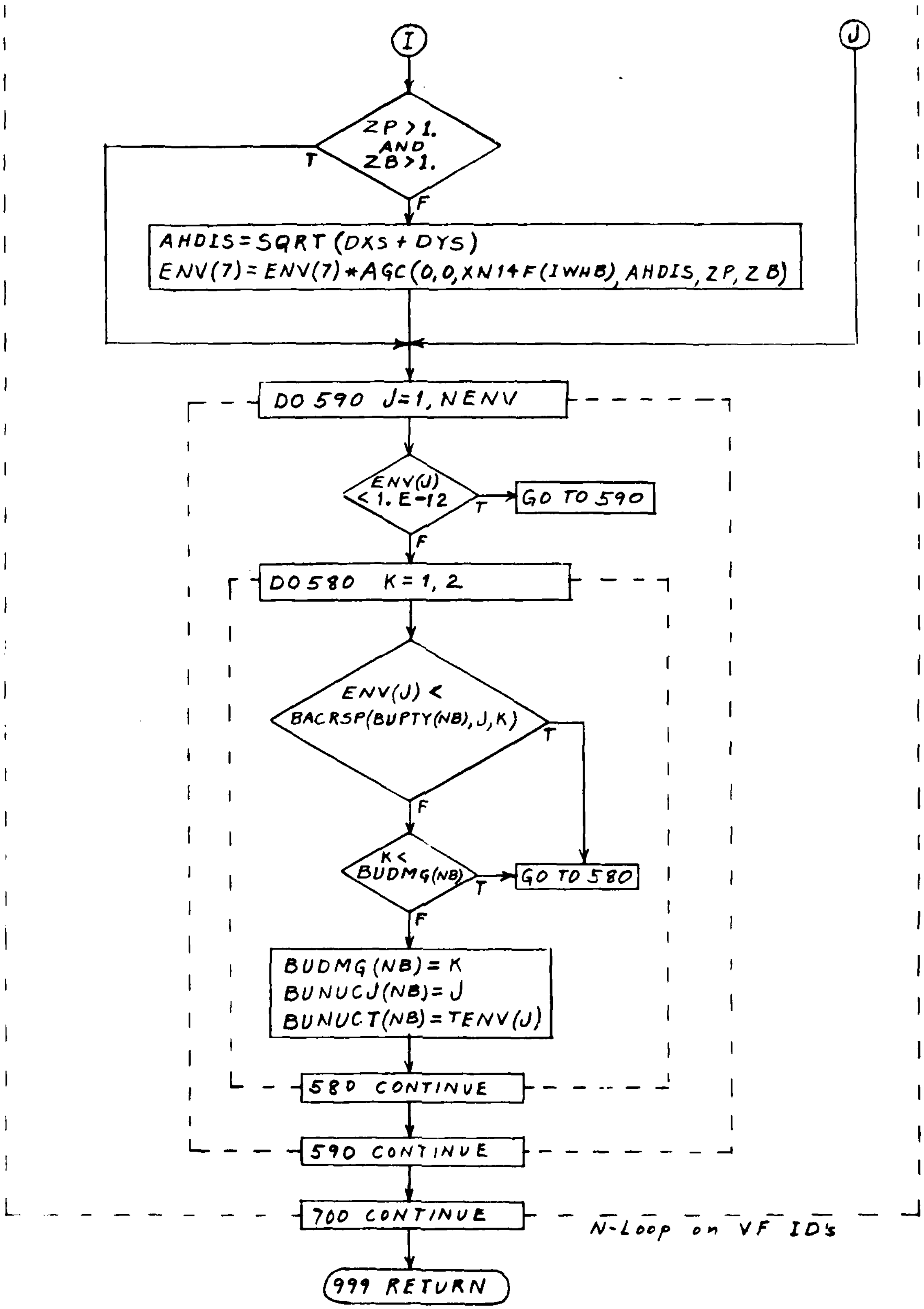


Figure 11-4 Continued. Subroutine NUCLER (Page 7 of 7)

SUBROUTINES REQUIRED

The following list of FORTRAN subroutines is required to support the operation of NUCLER:

BLAST	Computes blast overpressure and particle velocity, and calls other routines to return the five blast-related environment intensities and times.
BLINT	Computes the time and position of the intercept between the shock front and a moving target.
FENV	Computes the radiation intensities - neutron fluence, prompt gamma peak dose rate, and thermal fluence.
MATM62	A standard atmosphere model that returns pressure, temperature, sound speed, and density as a function of altitude.
MOVEL	Updates the positons of all the Red and Blue units in the game.
PULSE	Computes overpressure impulse at low altitudes.
SCALE	Computes shock scaling factors to permit fitting to standard curves for 1-KT sea level data.
VPARTS	Resolves particle velocities into axial and transverse components for moving targets.

In addition to the foregoing subroutines, NUCLER also requires the use of the following FORTRAN function subprograms:

AGC	Air-ground correction factor for neutron fluence at low altitude.
PFA	Free air overpressure.
PMS	Overpressure in the Mach stem region.
POLYNF	Polynomial expansion for interpolation.
RHOX	Burst-to-target air mass integral.
RSHK	Shock radius as a function of time.
RTPP	Triple point path.
RTRIR	Boundary between incident and reflected shock regions.

AD-A120 477 DESIGN NOTEBOOK FOR NAVAL AIR DEFENSE SIMULATION (NADS) (U) TRW DEFENSE SYSTEMS GROUP MCLEAN VA WATERWHEEL PROGRAM OFFICE R W COVEY ET AL. 15 SEP 82 4/4

UNCLASSIFIED N00014-81-C-0715 F/G 15/3 NL

END

FILMED

DTIC

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TERPL Data interpolation for RHOX and FENV.

TERPL2 Data interpolation for PULSE.

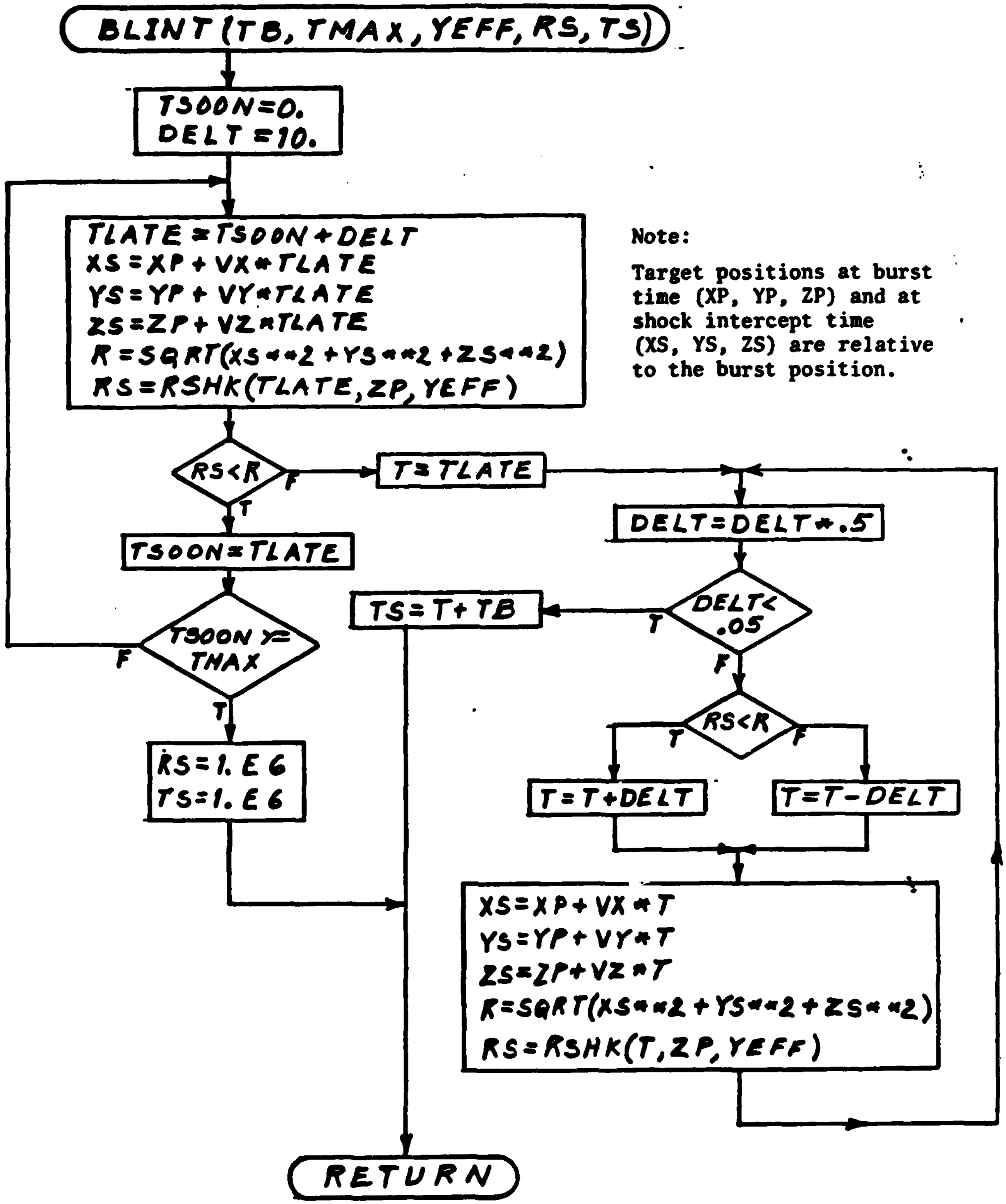
TERPT Data interpolation for PMS, RTPP, and RTRIR.

TSHK Shock arrival time as a function of yield, range, and altitude.

Subroutine BLINT (Figure 11-5)

Subroutine BLINT is called by NUCLER to determine where a moving target will intercept the shock front, if at all, and when. Because the shock expands at a rate that is quite non-linear and does not follow a simply written equation, an iterative process is used. An input TMAX is a time interval that corresponds with the maximum range of potential damage, i.e., if the shock arrives more than TMAX seconds after the burst it will be too weak to be of interest.

To reduce the number of iterations, the TMAX is first sampled at 10-second intervals to find the first one with the radius to the target, R, less than the shock radius, RS. If none is found, it is assumed that the target never encounters a shock of significant intensity. Function RSHK is called to compute RS after the target's radius is updated to each sample time. If an encounter is found, then the iteration proceeds by successively dividing the 10-second DELT into smaller increments and shifting the tentative time, T, up or down according as the last try was too small or too large. When DELT has been divided to less than 0.05 seconds, BLINT returns, having added the latest value of T to the burst time, TB, to get the shock intercept time, TS, and updating common block ENV with the corresponding intercept position, XS, YS, and ZS.



Note:

Target positions at burst time (XP, YP, ZP) and at shock intercept time (XS, YS, ZS) are relative to the burst position.

Figure 11-5. Subroutine BLINT

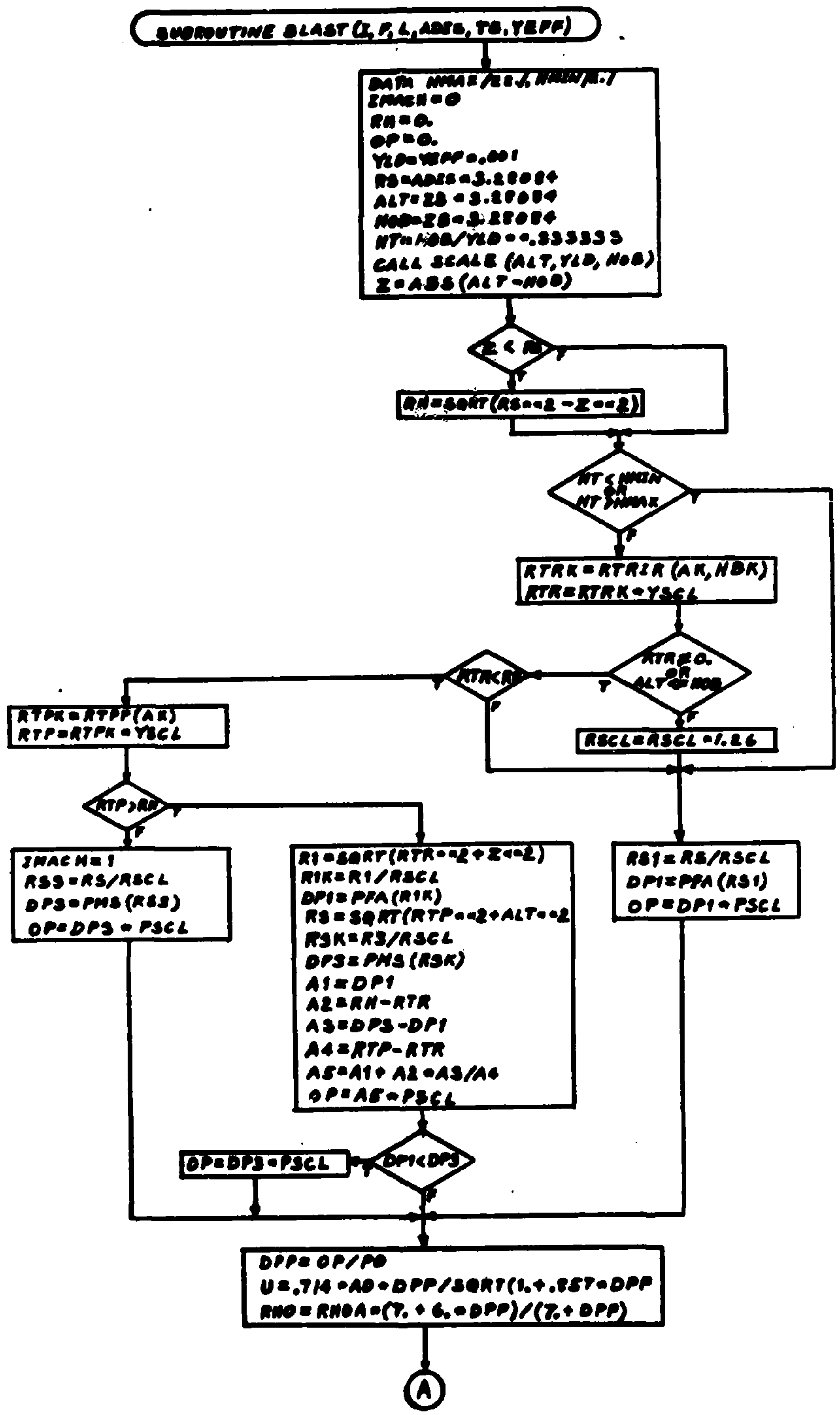


Figure 11-6. Subroutine BLAST

(Page 1 of 2)

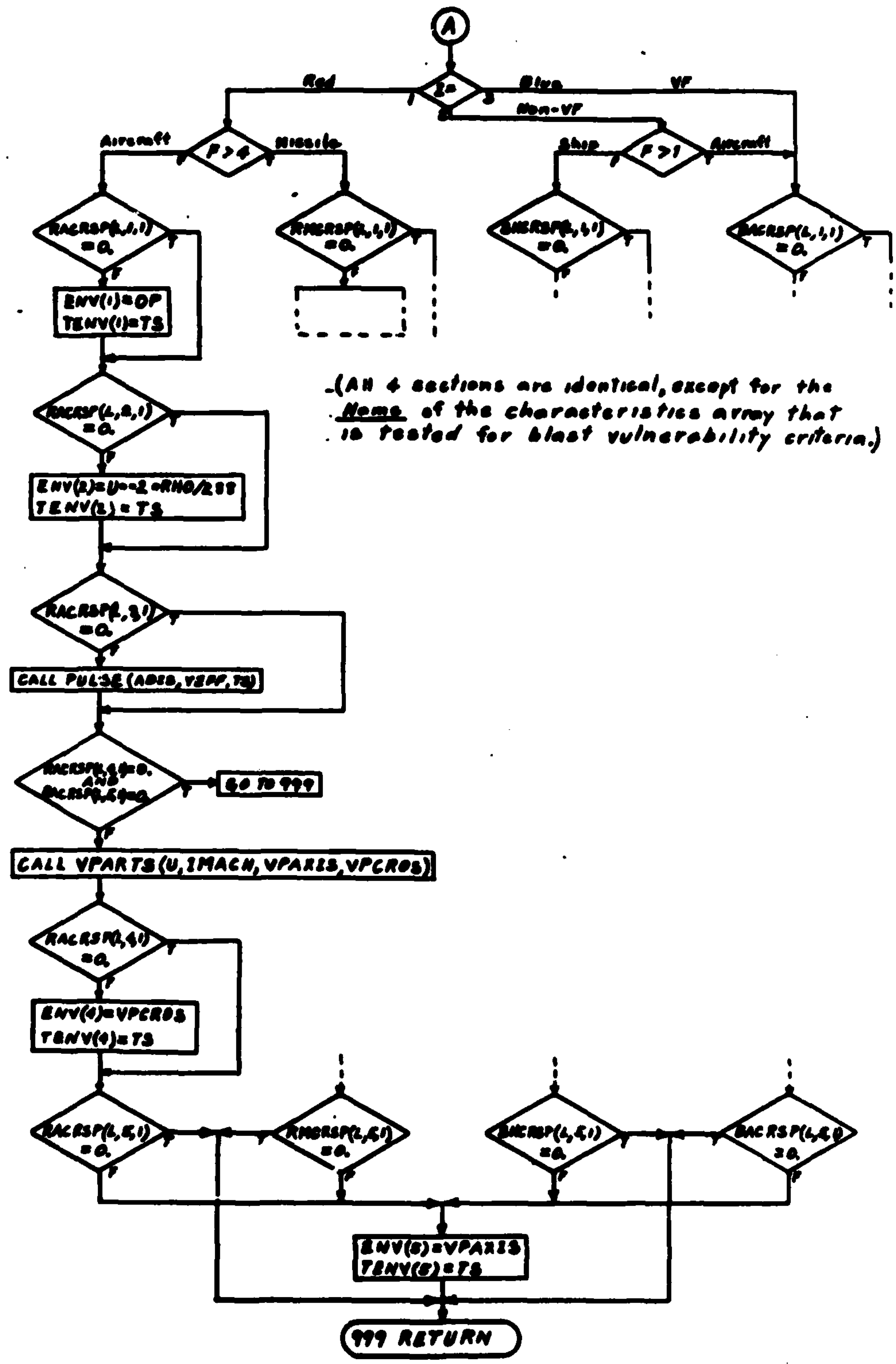


Figure 11-6 Continued. Subroutine BLAST

(Page 2 of 2)

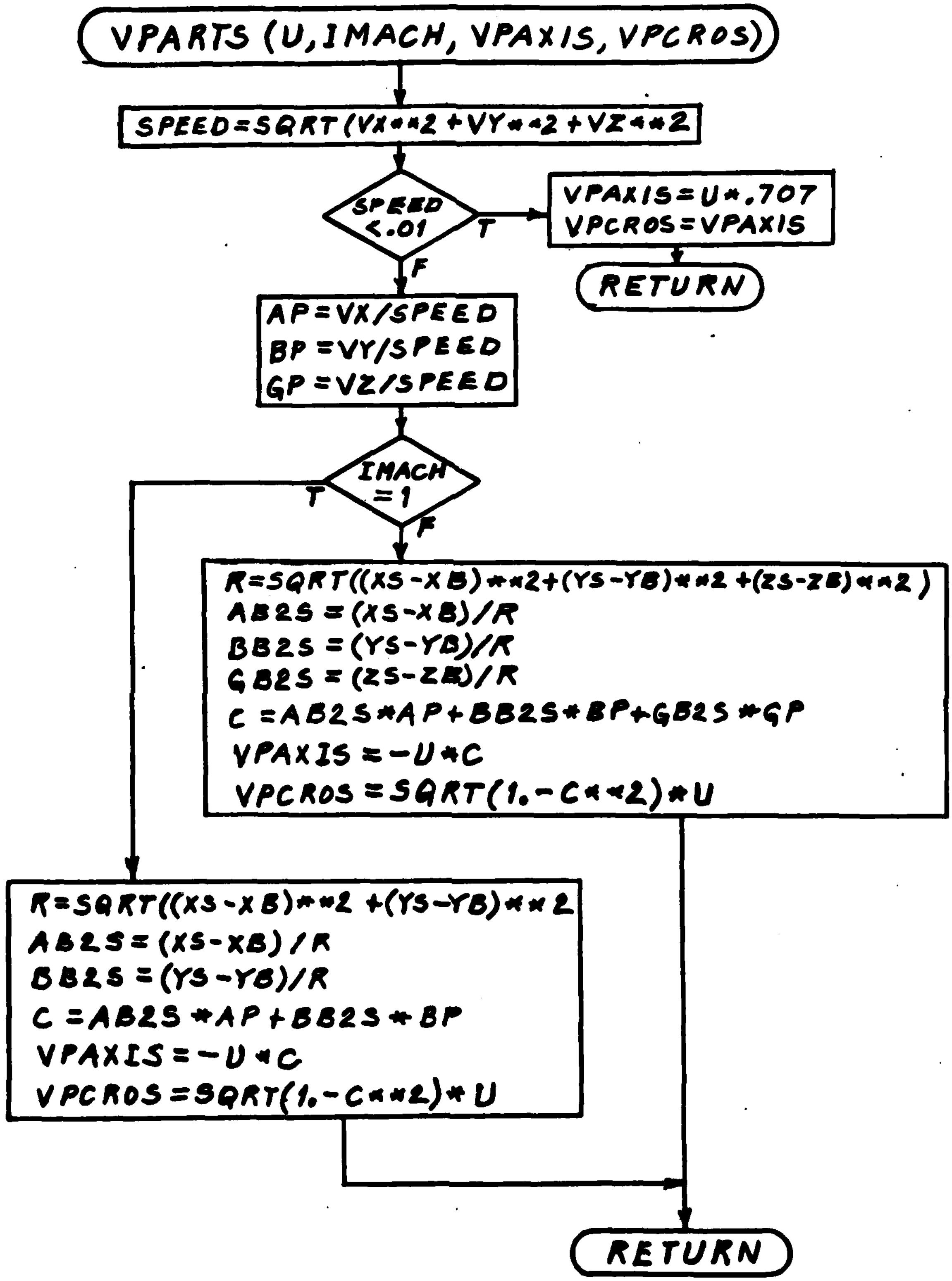


Figure 11-7. Subroutine VPARTS

11.4 NUCLEAR FIREBALL MASKING

One of the phenomena associated with nuclear fireballs is that their intense ionization renders them opaque to radio transmissions for a substantial time after the burst. Line-of-sight communication paths and radar paths are completely cut off if they pass through a fireball during its effective lifetime.

To simulate that effect, NADS evaluates the effective lifetime of each burst, and its size and position during that interval, to ascertain whether it intersects any particular line-of-sight.

The lifetime is based on an empirical equation that approximates the data in the EM-1 manual for a restricted set of conditions, viz. that the burst altitude be less than 20 km for communication frequencies or less than about 30 km for radar frequencies. The approximation is

$$T = (40 + 4.5\sqrt{HOB})\ YLD^{0.08}$$

where T is effective lifetime, in seconds
HOB is height of burst, in kilometers
YLD is yield, in kilotons.

This computation is done in subroutine NUCLER at the time of each burst. The decay time for each burst is stored in the NUCLOG common block, along with the burst time and burst position.

Subroutine FIRBAL is called at the time of a message transmission between two known positions. FIRBAL scans the list of decay times to see if any fireballs are currently active. If so, it computes the present height and calls subroutine DLOS to determine the fireball's distance from the line-of-sight. FIRBAL then computes the fireball radius to determine whether masking exists. If it does, the message transmission event is cancelled instead of executed.

The fireball height is computed as

$$H_{fb} = HOB + H_n H_r \quad , \text{ where}$$

$$H_n = 1 - \exp(-t/160) \quad , \text{ and}$$

$$H_r = 10 + \frac{YLD^{0.31}}{12.8}(HOB + 10) \quad .$$

The fireball radius is computed in three segments. The initial radius (essentially instantaneous at the time of burst) is an empirical approximation of the curves given in EM-1:

$$R_o = .034 \times 2^{LogYLD} \times 2^{HOB/20} \; .$$

The growth during the next six seconds is assumed to be at a rate that is decaying exponentially to a final rate that is constant from six seconds until the end of the fireball's masking lifetime (at most, a few minutes). The six-second exponential growth is

$$R_t = A(1-\exp(-t/B)) \quad , \text{ for } t \le 6, \text{ where}$$

$$A = R_o(2.77-.29R_o)$$

$$B = 2.45 + .25\, R_o$$

The final constant rate growth is

$$R_2 = 0.1\, R_o(t-6) \quad , \text{ for } t>6.$$

This treatment ignores the transition from spherical form to toroidal, because the small gain in accuracy would greatly complicate the geometry.

NADS DISTRIBUTION LIST

	No. of Copies
Office of the Chief of Naval Operations Department of the Navy Washington, D.C. 20350	
Attn: OP-654D	2
OP-009	1
OP-05	1
OP-96	1
OP-981	1
OP-950	1
OP-955	1
Naval Sea Systems Command Department of the Navy Washington, D.C. 20362	
Attn: NSEA-003	1
NSEA-06A	1
Naval Air Systems Command Department of the Navy Washington, D.C. 20360	
Attn: NAIR-03	1
Theater Nuclear Warfare Project Office Department of the Navy Washington, D.C. 20360	2
Center for War Gaming Naval War College Newport, Rhode Island 02840	1
U.S. Naval Surface Weapons Center White Oak Laboratory Silver Spring, MD 20910	
Attn: Technical Library & Information Services Branch	1
R-14	1
F-30	1
Commander, Naval Weapons Center China Lake, CA 93555	
Attn: A. Fouse, Code 32607	1
Commander, Naval Ocean Systems Center San Diego, CA 92152	
Attn: 8321	1

NADS DISTRIBUTION LIST (continued)

	No. of Copies
Commander, Training Command Pacific San Diego, CA 92147	1
Commander, Third Fleet Pearl Harbor, Hawaii Attn: N7	1
U.S. Naval Postgraduate School Monterey, CA 93940 Attn: Center for War Gaming Dept. of Operations Research For: Prof. R.N. Forrest	1
Center for Naval Analyses 2000 N. Beauregard Alexandria, VA 22311 Attn: Document Center	1
Director Defense Nuclear Agency Washington, D.C. 20305 Attn: NATA	1
NATD	1
Technical Library Division	1
Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209 Attn: Tactical Technology Office	1
Studies Analysis and Gaming Agency The Joint Chiefs of Staff Washington, D.C. 20301	1
Joint Strategic Target Planning Staff Building 500 Offutt Air Force Base, Nebr. 68113 Attn: War Gaming Divsion	1
Defense Technical Information Center Cameron Station, Bldg. 5 Alexandria, VA 22314	2

The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, Maryland 20810
Attn: Ray Guenther 3

Institute for Defense Analysis
1801 N. Beauregard St.
Alexandria, VA 22311
Attn: Documents Acquisition 1

Kaman Sciences Corporation
P.O. Box 7463
Colorado Springs, CO 80933
Attn: M. Brosee, KSC Washington Operations 1
W.J. Rudy 1

Ketron, Inc.
Rosslyn Center, 18th Floor
1700 N. Moore
Arlington, VA 22209 1